ELSEVIER

Discrete Applied Mathematics 123 (2002) 257–302

DISCRETE APPLIED MATHEMATICS

Selected topics on assignment problems ☆

Rainer E. Burkard

Technische Universität Graz, Institut für Mathematik, Steyrergasse 30, A-8010 Graz, Austria

Received 2 December 1999; received in revised form 4 April 2000; accepted 10 April 2000

Abstract

We survey recent developments in the fields of bipartite matchings, linear sum assignment and bottleneck assignment problems and applications, multidimensional assignment problems, quadratic assignment problems, in particular lower bounds, special cases and asymptotic results, biquadratic and communication assignment problems. © 2002 Elsevier Science B.V. All rights reserved.

MSC: 90C27; 90B80; 68Q25; 90C05

Keywords: Assignment problem; Bottleneck assignment problem; Multidimensional assignment problem; Quadratic assignment problem; Special cases; Asymptotic theory; Quadratic bottleneck assignment problem; Biquadratic assignment problem; Communication assignment problem

1. Introduction

Assignment problems deal with the question how to assign n items (jobs, students) to n other items (machines, tasks). Their underlying combinatorial structure is an *assignment,* which is nothing else than a bijective mapping φ between two finite sets of n elements. In the optimization problem we are looking for the *best* assignment, i.e., we have to optimize some objective function which depends on the assignment φ. Assignments can be represented in different ways. The bijective mapping between two finite sets V and W can be represented in a straight forward way by a *perfect matching* in a bipartite graph $G=(V,W;E)$, where the vertex sets V and W have n vertices. Edge $(i,j)\in E$ is an edge of the perfect matching iff $j=\varphi(i)$, cf. Fig. 1

E-mail address: burkard@opt.math.tu-graz.ac.at (R.E. Burkard).

☆ This research has been supported by Spezialforschungsbereich F 003 "Optimierung und Kontrolle", Projektbereich Diskrete Optimierung.

PII: S0166-218X(01)00343-2

$$\varphi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix}$$

$$X_\varphi = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

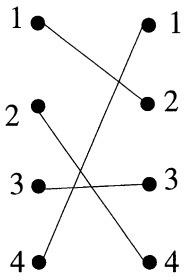

Fig. 1. Different representations of assignments.

By identifying the sets V and W we get the representation of an assignment by a *permutation*. Every permutation φ of the set $N = \{1, \ldots, n\}$ corresponds in a unique way to a *permutation matrix* $X_\varphi = (x_{ij})$ with $x_{ij} = 1$ for $j = \varphi(i)$ and $x_{ij} = 0$ for $j \neq \varphi(i)$. This matrix X_φ can be viewed as adjacency matrix of the bipartite graph G representing the perfect matching, see Fig. 1.

The set of all assignments (permutations) of n items will be denoted by $\mathscr{S}_n$ and has $n!$ elements. We can describe this set by the following constraints called *assignment constraints*.

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \text{for all } j = 1, \ldots, n,$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{for } i = 1, \ldots, n,$$

$$x_{ij} \in \{0, 1\} \quad \text{for all } i, j = 1, \ldots, n. \tag{1}$$

The set of all matrices $X = (x_{ij})$ fulfilling the assignment constraints will be denoted by $\mathbf{X}_n$.

When we replace the conditions $x_{ij} \in \{0, 1\}$ in (1) by $x_{ij} \geqslant 0$, we get a *doubly stochastic matrix*. The set of all doubly stochastic matrices forms the *assignment polytope* P_A. Birkhoff [15] showed that the assignments correspond uniquely to the vertices of P_A. Thus every doubly stochastic matrix can be written as convex combination of permutation matrices.

Theorem 1.1 (Birkhoff [15]). *The vertices of the assignment polytope correspond uniquely to permutation matrices.*

Flows in networks offer another model to describe assignments. Let $G = (V, W; E)$ be a bipartite graph with $|V| = |W| = n$. We embed G in the network $\mathscr{N} = (N, A, c)$ with *node set* N, *arc set* A and *arc capacities* c. The node set N consists of a *source* s, a *sink* t and the vertices of $V \cup W$. The source is connected to every node in V by a directed arc of capacity 1, every node in W is connected to the sink by a directed

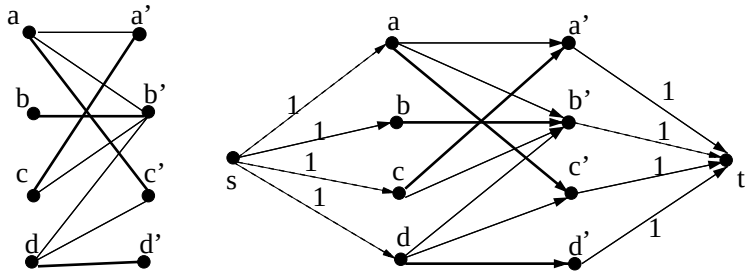

Fig. 2. Perfect matching in a bipartite graph and corresponding network flow model. A minimum cut is given by $\{s, b, b'\}$. The dashed arcs lie in the cut. Thus the cut has value 4.

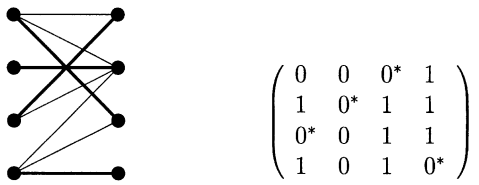

Fig. 3. 0–1 matrix model of a bipartite graph and the corresponding minimum vertex cover which corresponds to the cut in Fig. 2. An assignment is given by the entries marked by $*$.

arc of capacity 1, and every arc in E is directed from V to W and supplied with an infinite capacity. The *maximum network flow problem* asks for a flow with maximum value $z(f)$. Obviously, a maximum integral flow in the special network constructed above corresponds to a matching with maximum cardinality, see Fig. 2. A *cut* in the network $\mathcal{N}$ is a subset C of the node set N with $s \in C$ and $t \notin C$. The value $u(C)$ of cut C is defined as (Fig. 3)

$$u(C) := \sum_{\substack{x \in C,\, y \notin C \\ (x,y) \in A}} c(x, y),$$

where $c(u, v)$ is the capacity of the arc (x, y).

Ford and Fulkerson's famous *Max Flow-Min Cut Theorem* [69] states that the value of a maximum flow equals the minimum value of a cut. This max flow-min cut theorem can directly be translated in König's Matching Theorem [99]. Given a bipartite graph G, a *vertex cover* (cut) in G is a subset of its vertices such that every edge is incident with at least one vertex in this set.

Theorem 1.2 (König's Matching Theorem [99]). *In a bipartite graph the minimum number of vertices in a vertex cover equals the maximum cardinality of a matching.*

Let us now formulate this theorem in the language of 0–1 matrices. Given a bipartite graph $G = (V, W; E)$ with $|V| = |W| = n$, we define the zero-adjacency matrix B of G

as $(n \times n)$ matrix $B = (b_{ij})$ where

$$b_{ij} := \begin{cases} 0 \text{ if } (i,j) \in E, \\ 1 \text{ if } (i,j) \notin E. \end{cases}$$

A *zero-cover* is a subset of the rows and columns of matrix B which contains all 0 elements. A row (column) which is an element of a zero-cover is called a *covered row* (*covered column*). Now we get

Theorem 1.3. *There exists an assignment φ with $b_{i\varphi(i)} = 0$ for all $i = 1, \ldots, n$, if and only if the minimum zero cover has n elements.*

Since a maximum matching corresponds uniquely to a maximum flow in the corresponding network $\mathcal{N}$, we can construct a zero-cover in the zero-adjacency matrix B by means of a minimum cut C in this network: if node $i \in V$ of the network does not belong to the cut C, then row i is an element of the zero-cover. Analogously, if node $j \in W$ belongs to the cut C, then column j is an element of the zero-cover.

2. Perfect matchings

In this section we deal with the question, whether there exists an assignment (i.e. a perfect matching) in a given bipartite graph or not. A basic answer to this question is provided by Hall's Marriage Theorem [83]. For a vertex $v \in V$ let $N(v)$ be the set of its neighbors, i.e., the set of all vertices $w \in W$ which are connected with v by an edge in E. Thus $N(v)$ contains the "friends" of v. Moreover, for any subset V' of V let $N(V') = \bigcup_{v \in V'} N(v)$.

Theorem 2.1 (Marriage Theorem [83]). *Let $G = (V, W; E)$ be a bipartite graph with $|V| = |W|$. G contains an assignment (perfect matching, marriage) if and only if for all subsets V' of V:*

$$|V'| \leqslant |N(V')| \quad \text{(Hall condition)}.$$

When we want to apply this theorem for a special graph we have to check exponentially many subsets V' of V. Hopcroft and Karp [88] gave a polynomial-time algorithm to decide this question. They construct a perfect matching in $O(|E|\sqrt{|V|})$ steps, if it exists. This is done by a careful analysis of an algorithm for finding a maximum flow in a network with arc capacities 1.

Alt et al. [5] improve the complexity for dense graphs by a fast matrix scanning technique and obtain an $O(|V|^{1.5}\sqrt{|E|/\log|V|})$ implementation for the Hopcroft–Karp algorithm. They save the factor $\log|V|$ by storing the occurring 0–1 matrices (e.g. the adjacency matrix of the graph) in blocks of length $\log|V|$ as RAM-words which can be processed in constant time.

A randomized algorithm can decide even faster, whether G contains an assignment or not. This algorithm is based on the following theorem of Tutte [148]. The *Tutte matrix* $A(x) = (x_{ij})$ of an (undirected) graph $G = (V, E)$ is a skew symmetric matrix

$$\begin{pmatrix} 0 & 0 & 0 & x_{14} & x_{15} & x_{16} \\ 0 & 0 & 0 & x_{24} & 0 & 0 \\ 0 & 0 & 0 & x_{34} & 0 & 0 \\ -x_{14} & -x_{24} & -x_{34} & 0 & 0 & 0 \\ -x_{15} & 0 & 0 & 0 & 0 & 0 \\ -x_{16} & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

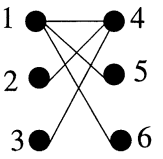

Fig. 4. The Tutte matrix of a bipartite graph.

with indeterminate entries x_{ij}, $x_{ij} = -x_{ji}$, where $x_{ij} = x_{ji} \equiv 0$, iff (i,j) is not an edge of G, see Fig. 4.

Theorem 2.2 (Tutte [148]). *Let $A(G)$ be the Tutte matrix of graph $G=(V,E)$. There exists a perfect matching in G, if and only if the determinant of $A(G)$ is not identically equal to 0.*

This theorem can now be used in the following algorithm.

Algorithm.

1. Generate randomly the values of $x_{ij}, 1 \leqslant i,\ j \leqslant n$ from the set $\{1,2,\ldots,|E|^2\}$.
2. Compute $\det A(G)$.
3. If $\det A(G) \neq 0$, stop. Graph G contains an assignment. Otherwise goto Step 1, unless a prespecified number r of repetitions is already reached.

Due to (Coppersmith and Vinograd [53]) the determinant in Step 2 can be computed in $O(n^{2.376})$ steps. The algorithm errs, if it yields the value 0 for the determinant, but $\det A(G)$ is not identically equal to 0. This happens if the random numbers generated in Step 1 hit by chance a root of the polynomial $\det A(G)$. According to a result of Schwarz [142] this happens with a probability equal to $1/|E|$. Thus the algorithm errs after r repetitions with a probability $(1/|E|)^r$.

This algorithm is faster than the best known deterministic algorithm. Note, however, that this procedure does not provide an assignment explicitly. A similar parallel algorithm which also provides an assignment, is due to Mulmuley et al. [115]. Their algorithm requires $O(|V|^{3.5}|E|)$ processors to find a perfect matching in $O(\log^2|V|)$ time.

3. The time slot assignment problem

As an application of perfect matchings we consider the following problem from telecommunication. For remitting data from one sending earth station via a satellite to another receiving earth station, the so-called *time division multiple access* (*TDMA*) *technique* may be used. This technique buffers first the data to be remitted in the ground stations. Then they are sent in very short data bursts to the satellite. There they are received by transponders and again transmitted to the earth, namely to the receiving earth stations. A *transponder* connects one sending station with a receiving station. For a fixed time interval of variable length λ_k the n sending stations are

connected with the receiving stations via the n transponders onboard the satellite, i.e., a certain *switch mode* is applied. Mathematically, a switch mode P_k corresponds to a permutation matrix, whose 1-entries show the current connections. After a short while the connections onboard the satellite are simultaneously changed and in the next time interval new pairs of sending and receiving stations are connected via transponders, i.e., a new switch mode is applied.

The *time slot assignment problem* tackles the questions as to which switch modes should be applied and how long each of them lasts such that a given amount of data can be remitted in the shortest possible time. Given an $(n \times n)$ traffic matrix $T=(t_{ij})$, where t_{ij} describes the amount of information to be remitted from the ith sending station to the jth receiving station, we have to determine the switch modes P_k, $k = 1, 2, \ldots$, and the nonnegative lengths λ_k of the corresponding time slots during which the switch modes P_k are applied, such that all data are remitted in the shortest possible time. This leads to the following mathematical model:

$$\begin{aligned} \min \quad & \sum_k \lambda_k \\ \text{s.t.} \quad & \\ & \sum_k \lambda_k p_{ij}^{(k)} \geqslant t_{ij}, \quad \text{for } 1 \leqslant i,\ j \leqslant n, \\ & \lambda_k \geqslant 0, \quad \text{for all } k. \end{aligned} \tag{2}$$

This problem can be solved optimally by the following algorithm of complexity $O(n^4)$, see e.g. [19]. First we assume that the traffic matrix T has constant row and column sums. Otherwise, let t^* be the maximum value of the row and column sums. Now we can fill up the matrix in a straightforward way by increasing some elements such that all row and column sums are equal to t^*. Since due to the Theorem of Birkhoff a doubly stochastic matrix is the convex combination of permutation matrices, we can write the traffic matrix T as a weighted sum of switch modes. The sum of weights equals t^*, i.e., this decomposition is optimal, since no two elements of the same row or column can be remitted at the same time.

Algorithm (Matrix decomposition).

1. Let $k := 1$.
2. Construct a bipartite graph with $|V| = |W| = n$ and the following edges:
   $(i, j) \in E$ iff $t_{ij} > 0$.
3. Find a perfect matching φ_k corresponding to a switch mode P_k in this graph.
4. Let $\lambda_k := \min\{t_{i\varphi(i)}\}$.
5. Form $T := T - \lambda_k P_k$.
6. If $T \neq \mathbf{0}$, set $k := k + 1$ and goto 2; otherwise stop.

Note that in every iteration we have a matrix T with constant row and column sums. Therefore there exists a perfect matching in Step 2 and it can be found by Hopcroft and Karp's procedure. We also know from the dimension $(n-1)^2$ of the assignment polytope that T is decomposed in at most $n^2 - 2n + 2$ different switch

modes. Now we can question, whether this mathematically optimal procedure leads to a technically feasible solution. Onboard the satellite are about 40 transponders. A time interval which has to be split up into different time slots has a length of about 2 ms. But it is technically infeasible to switch about 1600 times within 2 ms from one mode to another. Therefore one has to restrict the number of switch modes. But then, unfortunately, the time slot assignment problem with the additional restriction that the number of switch modes is $O(n)$, becomes $\mathcal{NP}$-hard, see [137]. Thus one has to solve the constraint time slot assignment problem by heuristics, see e.g. the comments on balanced linear assignment problems at the end of Section 4.3.

Another possibility consists in considering a different model. According to a proposal of Lewandowski et al. [106] a system of $2n$ switch modes $P_1,\dots,P_n$, $Q_1,\dots,Q_n$ could be fixed onboard the satellite, where

$$\sum_k P_k = \sum_l Q_l = \mathbf{1}$$

and $\mathbf{1}$ is the matrix with 1-entries only. For such a setting the time slot assignment problem can be transformed in an ordinary linear sum assignment problem. For details see [19].

4. Linear assignment problems

4.1. A general solution method for algebraic linear assignment problems

Linear assignment problems can be solved by only *adding*, *subtracting* and *comparing* the cost coefficients. A careful reasoning shows moreover that a subtraction $b-a$ occurs only in the case that the cost element a is not greater than b. Thus, when we want to determine $b-a$, we ask for an element c such that $a+c=b$. These considerations lead to an algebraic model, originally introduced in Burkard et al. [36], which allows to formulate and solve linear assignment problems within a general framework. The underlying algebraic structure is a so-called *d-monoid*, i.e., a totally ordered commutative semigroup $(H,*,\leqslant)$ with composition $*$ and linear order relation $\leqslant$, which fulfills in addition the following axiom

$$\text{If } a \leqslant b, \quad \text{then there exists an element } c \in H \text{ such that } a * c = b. \tag{3}$$

Given n^2 cost coefficients $c_{ij} \in H$, the *algebraic linear assignment problem* can be formulated as

$$\min_{\varphi} (c_{1\varphi(1)} * c_{2\varphi(2)} * \cdots * c_{n\varphi(n)}). \tag{4}$$

It is possible to solve algebraic assignment problems in $O(n^4)$ steps. But if we additionally assume that the *weak cancellation rule*

$$\text{If } a * c = b * c, \quad \text{then either } a = b \text{ or } a * c = b \tag{5}$$

holds in $(H, *, \preccurlyeq)$, we can solve these problems in $O(n^3)$ steps, see [43]. For further results in this direction, consult the survey on algebraic optimization by Burkard and Zimmermann [44].

Special examples for d-monoids which obey the weak cancellation rule are:

- $H = \mathbb{R}$ with addition as composition and the usual order relation. This model leads to linear sum assignment problems (LSAP):

$$\min_{\varphi} (c_{1\varphi(1)} + c_{2\varphi(2)} + \cdots + c_{n\varphi(n)}).$$

- H is the set of extended real numbers $\bar{\mathbb{R}}$ (including $-\infty$) with the usual order relation. The composition is defined by $a * b := \max(a, b)$. This model leads to linear bottleneck assignment problems:

$$\min_{\varphi} \max\{c_{1\varphi(1)}, c_{2\varphi(2)}, \ldots, c_{n\varphi(n)}\}.$$

- $H = \mathbb{R}^n$, the composition is the vector addition and the order relation is the lexicographical order. This leads to lexicographical sum assignment problems.
- $H = \bar{\mathbb{R}} \times \mathbb{R}$, $\preccurlyeq$ is the lexicographical order and the composition is defined by

$$(a, b) * (c, d) := \begin{cases} (a, b) & \text{if } a \geqslant c, \\ (a, b + d) & \text{if } a = c. \end{cases}$$

  This leads to the so-called *time–cost assignment problem*. Let a denote a time and let b denote the corresponding cost. In a time cost assignment problem we want to find an assignment which first minimizes the maximum occurring time. Secondly, under all solutions which yield this time, a solution with minimum cost is to be found. Such problems occur if n customers have to be served as fast a possible, e.g. under emergency aspects, and then a cost minimal optimal solution should be found.

Linear (algebraic) assignment problems can be solved by transforming the $n \times n$ cost matrix $C = (c_{ij})$ until we find a zero cover which has n elements (cf. Theorem 1.3). Thus we have to define *zero elements* in H, we have to describe the transformations of the cost matrix C which we shall call *admissible transformations*, and we have to explain in which way these admissible transformations should be applied in order to get finally a minimum zero cover.

Let us start with the definition of admissible transformations. We abbreviate the objective function value of permutation φ with respect to the cost matrix C by

$$z[C, \varphi] := c_{1\varphi(1)} * c_{2\varphi(2)} * \cdots * c_{n\varphi(n)}.$$

Definition 4.1 (Admissible transformation). A transformation T of the $n \times n$ matrix $C = (c_{ij})$ to the matrix $\bar{C} = (\bar{c}_{ij})$ is called *admissible* with *index* $z(T)$, if

$$z[C, \varphi] = z(T) * z[\bar{C}, \varphi]$$

for all $\varphi \in \mathscr{S}_n$.

If we perform an admissible transformations T after an admissible transformation S, we get again an admissible transformation. If S and T have the indices $z(S)$ and $z(T)$, respectively, their composition has index $z(S) * z(T)$. The definition above states a property of admissible transformations, but does not explain, how we can get it. This is provided by the following theorem:

Theorem 4.2 (Admissible transformations for assignment problems [36]). *Let* $I,J \subseteq \{1,2,\ldots,n\}$, $m := |I| + |J| - n \geqslant 1$, *and* $c := \min\{c_{ij}: i \in I, j \in J\}$. *Then the transformation* $C \mapsto \bar{C}$ *defined by*

$$\bar{c}_{ij} * c = c_{ij}, \quad \text{for } i \in I,\ j \in J$$

$$\bar{c}_{ij} = c_{ij} * c, \quad \text{for } i \notin I,\ j \notin J$$

$$\bar{c}_{ij} = c_{ij}, \quad \text{otherwise}$$

is admissible with $z(T) = c * c * \cdots * c$, *where the expression on the right-hand side contains* m *factors.*

Note that we make use of (3) in the first line of the definition of $\bar{c}_{ij}$!

In the semigroup H the role of 0-elements is replaced by so-called *dominated elements.* An element $a \in H$ is dominated by an element $z \in H$, if $a * z = z$. Thus in $(\mathbb{R}, \leqslant, +)$ the 0 is dominated by any other number. Now we can formulate the following optimality criterion:

Theorem 4.3. *Let* $T : C \to \bar{C}$ *be an admissible transformation such that there exists a permutation* $\hat{\varphi}$ *with the following properties*:
1. $z(T) * \bar{c}_{ij} \geqslant z(T)$,
2. $z[\bar{C}, \hat{\varphi}] * z(T) = z(T)$.

Then $\hat{\varphi}$ *is an optimal assignment with value* $z(T)$.

The first property in Theorem 4.3 says that all cost coefficients $\bar{c}_{ij}$ are "non-negative" (with respect to $z(T)$). The second property of Theorem 4.3 says that the current objective function value is already dominated by $z(T)$, i.e., has value "0".

Proof. Let φ be an arbitrary permutation. According to Definition 4.1 and properties (1) and (2) of the proposition above we get:

$$z[C, \varphi] = z(T) * z[\bar{C}, \varphi] \geqslant z(T) = z(T) * z[\bar{C}, \hat{\varphi}] = z[C, \hat{\varphi}].$$

Therefore $\hat{\varphi}$ is optimal. □

Now we have to specify in which way the admissible transformations should be applied. This is stated in the following algorithm:

4.1.1. Algorithm for solving linear algebraic assignment problems

1. Perform *row reductions* in matrix C, i.e., perform admissible transformations with $I = \{k\}$, $J = \{1,2,\ldots,n\}$. Start with $k = 1$ and let $z := z(T)$ be the corresponding

index. Continue with $k = 2, \ldots, n$ and update $z := z * z(T)$. Afterwards all elements in the transformed matrix are "nonnegative" with respect to z, namely $\bar{c}_{ij} * z \geqslant z$.
2. Perform *column reductions*, i.e., perform admissible transformations with $I = \{1, 2, \ldots, n\}$, $J = \{k\}$, for $k = 1, 2, \ldots, n$. Afterwards every row and column in the transformed cost matrix contains at least one element which is dominated by z. All other elements remain non-negative with respect to z.
3. Determine a maximum matching in the following bipartite graph $G = (V, W; E)$, where V contains the row indices of the transformed cost matrix, W the column indices and $(i, j) \in E$, iff $\bar{c}_{ij} * z = z$.
4. If the maximum matching is perfect, then stop: the optimal solution is given by this matching and z is the optimal value of the objective function. Otherwise, go to Step 5.
5. Determine a minimum cover of the transformed cost coefficients dominated by z. This cover yields the new index sets I and J. I contains the indices of the uncovered rows, J contains the indices of uncovered columns.
6. Perform an admissible transformation determined by the new index sets I and J as in Theorem 4.2, update $z := z * z(T)$, and go to Step 3.

It is rather straightforward to show that this algorithm yields an optimal solution of the algebraic assignment problem after at most $n^2 - 2n + 3$ admissible transformations. If the composition $*$ is specialized to "+", the algorithm described above becomes a variant of the Hungarian method. If the composition is specialized to the *max* operation, then we obtain the bottleneck assignment problem and the above algorithm is a variant of the threshold method.

Finally we address a case where the solution of an algebraic assignment problem can be stated explicitly. We say a cost matrix $C = (c_{ij})$ fulfills the *algebraic Monge property*, if it fulfills the following conditions:

$$c_{ij} * c_{kl} \preccurlyeq c_{il} * c_{kj}, \quad \text{for } 1 \leqslant i < k \leqslant n, \ 1 \leqslant j < l \leqslant n. \tag{6}$$

We can show that the following theorem holds, see [87,38].

Theorem 4.4. *If the cost matrix C of an algebraic linear assignment problem fulfills the algebraic Monge property* (6), *then this assignment problem is solved by the identical permutation id, defined by $id(i) = (i)$ for all $i = 1, 2, \ldots, n$.*

The Monge property depends on the proper numbering of the rows and columns of the matrix. A matrix C is called a *permuted Monge matrix*, if there exists a pair of permutations (φ, ψ) such that the matrix $C^{(\varphi,\psi)} = (c_{\varphi(i)\psi(j)})$ obtained from C by permuting its rows according to φ and its columns according to ψ, is a Monge matrix, see [38]. The problem of recognizing permuted algebraic Monge matrices is rather subtle in the general case. It can be shown that this problem is NP-hard, if $n \geqslant 3$ and the ordered semigroup fulfills no additional property. It becomes, however, polynomially solvable, if for instance a weak cancellation rule (4.1) is fulfilled. For details see [38].

4.2. Linear sum assignment problems

Linear sum assignment problems (LSAP) belong to the classical problems of mathematical programming. They occur mainly as subproblems in more complex situations like the travelling salesman problem, vehicle routing problems, personnel assignments and similar problems from practice. An interesting application in railway systems is described by Neng [119], who considers the problem of assigning engines to trains due to traffic constraints and formulates this problem as a linear assignment problem.

A large number of algorithms, sequential and parallel, has been developed for the LSAP, e.g. primal-dual algorithms, simplex-like methods, cost operation algorithms, forest algorithms and relaxation approaches. For a survey on these methods and available computer programs see the recent article of Burkard and Çela [24] or the annotated bibliography of Dell'Amico and Martello [56]. It should be pointed out that nowadays it is possible to solve large scale dense LSAPs (with $n \approx 10^6$) within a couple of minutes, see [105].

Whereas $O(n^3)$ is the best worst case complexity for sequential linear sum assignment algorithms, an algorithm with expected running time of $O(n^2 \log n)$ was developed by Karp [94] in the case of independent and uniformly distributed cost coefficients c_{ij} in $[0,1]$. This algorithm is a special implementation of the classical shortest augmenting path algorithm. It uses priority queues to compute shortest augmenting paths in $O(n^2 \log n)$ time which yields a worst case time complexity of $O(n^3 \log n)$.

For very large problems there is a need for good and fast heuristics. Karp et al. [96] developed a fast heuristic which runs in $O(n \log n)$ time in the worst case and $O(n)$ expected time. In the case of uniformly distributed cost coefficients in $[0,1]$ it provides a solution whose value is, smaller than $3 + O(n^{-a})$, for some $a > 0$, with probability $1 - O(n^{-a})$. The basic idea is to construct a "cheap" sparse subgraph of the given graph. Then it is shown that this sparse subgraph contains a perfect matching with high probability. If the subgraph does not contain a perfect matching a solution for the original LSAP instance is determined in a greedy way.

It has already been pointed out at the end of Subsection 4.1 that the identical permutation is an optimal solution of an LSAP if its cost matrix fulfills a Monge condition. This remains even true, when the Monge condition is relaxed to the so-called *weak Monge property* (cf. [58])

$$c_{ii} + c_{kl} \leqslant c_{il} + c_{ki}, \quad \text{for } 1 \leqslant i < k \leqslant n,\ 1 \leqslant i < l \leqslant n. \tag{7}$$

Analogously, it can be shown that the permutation φ defined by $\varphi(i) = n - i + 1$ for all i, is an optimal solution of an LSAP with an *Anti-Monge cost matrix* C, i.e. a cost matrix $C = (c_{ij})$ fulfilling

$$c_{ij} + c_{kl} \geqslant c_{il} + c_{kj} \quad \text{for } 1 \leqslant i < k \leqslant n,\ 1 \leqslant j < l \leqslant n. \tag{8}$$

As was pointed out in the last subsection, Monge properties depend on the proper numbering of the rows and columns of the considered matrix. In the case of sum problems, Deĭneko and Filonenko [54] designed an $O(n^2)$ algorithm which decides, whether an $n \times n$ matrix C is a permuted Monge matrix. Moreover, if C is a permuted Monge matrix, the algorithm constructs the appropriate permutations φ, ψ for the rows

and the columns within this time bound. As a consequence, the LSAP with a permuted Monge cost matrix can be solved in $O(n^2)$ time. The reader is referred to Burkard et al. [38] for a detailed discussion of Monge properties, and a description of the algorithm of Deĭneko and Filonenko.

An important special case of an LSAP with a permuted Monge cost matrix arises if the cost coefficients have the form

$$c_{ij} = u_i v_j \quad \text{for all } i, j$$

with non-negative numbers u_i and v_j. Such an LSAP can simply be solved in $O(n \log n)$ time by ordering the elements u_i and v_j, see the following theorem on minimum and maximum scalar products:

Theorem 4.5 (Hardy et al. [85]). *Let* $0 \leqslant u_1 \leqslant \cdots \leqslant u_n$ *and* $0 \leqslant v_1 \leqslant \cdots \leqslant v_n$. *Then for any permutation* φ

$$\sum_{i=1}^{n} u_i v_{n+1-i} \leqslant \sum_{i=1}^{n} u_i v_{\varphi(i)} \leqslant \sum_{i=1}^{n} u_i v_i.$$

4.3. Linear bottleneck assignment problems

Linear bottleneck assignment problems (LBAP) have the form

$$\min_{\varphi} \max_{1 \leqslant i \leqslant n} c_{i\varphi(i)}. \tag{9}$$

They were introduced by Fulkerson et al. [74] and occur e.g. in connection with assigning jobs to parallel machines so as to minimize the latest completion time. Another application occurs in locating objects in space. Let us consider n objects which are detected by two sensors at geographically different sites. Each sensor measures the angle under which the object can be seen, i.e., it provides n lines, on which the objects lie. The location of every object is found by intersecting the appropriate lines. The pairing of the lines is modeled as follows: let c_{ij} be the smallest distance between the ith line from sensor 1 and the jth line from sensor 2. Due to small errors during the measurements, c_{ij} might even be greater than 0 if some object is determined by lines i and j. Solving an LBAP with cost matrix $C = (c_{ij})$ leads to very good results in practice (cf. [16] who used, however, linear sum assignment problems instead of the error-minimizing bottleneck problems). A similar technique can be used for tracking missiles in space. If their locations at two different times t_1 and t_2 are known, we compute the (squared) Euclidean distances between any pair of old and new locations and solve the corresponding linear bottleneck assignment problem in order to match the points in the right way.

Considering bottleneck assignment problems, Gross [78] proved the following min–max theorem which was a starting point of the theory on blocking systems, see [63].

Theorem 4.6 (Gross [78]). *Let* $N = \{1, 2, \ldots, n\}$ *and let* $\mathscr{S}_n$ *be the set of all permutations* φ *of* N. *Then the following min–max equality holds for an arbitrary* $n \times n$

matrix $C=(c_{ij})$ *with elements* c_{ij} *drawn from a totally ordered set*:

$$\min_{\varphi\in\mathscr{S}_n}\max_{i\in N} c_{i\varphi(i)} = \max_{\substack{I,J\subseteq N\\ |I|+|J|=n+1}}\min_{i\in I,j\in J} c_{ij}. \tag{10}$$

Note the relationship to Theorem 4.2, where we perform a transformation with $\min_{i\in I,j\in J} c_{ij}$. Indeed, the algorithm of Section 4.1 leads to so-called *threshold algorithms* for solving bottleneck assignment problems: A threshold algorithm alternates between two phases. In the first phase a cost element c^*_{ij}—the *threshold value*—is chosen and a matrix $\bar{C}$ is defined by

$$\bar{c}_{ij} := \begin{cases} 1 \text{ if } c_{ij} > c^*_{ij}, \\ 0 \text{ if } c_{ij} \leqslant c^*_{ij}. \end{cases}$$

In the second phase it is checked, whether the bipartite graph with zero-adjacency matrix $\bar{C}$ contains a perfect matching or not. The smallest value c^*_{ij} for which the corresponding bipartite graph contains a perfect matching is the optimum value of the LBAP (9).

There are several ways to implement such a threshold algorithm. One possibility is to order the cost elements increasingly and to apply a binary search in the first phase. This leads to an $O(T(n)\log n)$ algorithm, where $T(n)$ is the time complexity for checking the existence of a perfect matching.

Another possibility is to mimic the Hungarian method. We start with

$$c^* := \max_{i,j}\left(\min_i c_{ij},\ \min_j c_{ij}\right) \tag{11}$$

and grow bottleneck augmenting paths as long as the matrix $\bar{C}$ does not contain an assignment with objective function value equal to 0, see [122,76,59,45,57]. FORTRAN codes for this method can be found in the book by Burkard and Derigs [31] and in Carpaneto and Toth [45]. The implementations differ in the determination of a starting solution and in the applied data structures. One of the most efficient implementations is described in Derigs [57]. A thorough recent investigation on computational issues concerning LBAPs can be found in Pferschy [126]. Among others Pferschy proposes an implementation using sparse subgraphs.

An algorithm with the currently best (theoretical) time complexity is obtained by combining the threshold approach with augmenting paths. This idea goes back to Gabow and Tarjan [75], who designed an algorithm with worst time complexity $O(m\sqrt{n\log n})$ for the LBAP whose underlying bipartite graph G has $2n$ vertices and m edges. For dense graphs this bound has been improved further by Punnen and Nair [133]. According to these authors we first solve

$$\min_{M\in F^*}\max_{(i,j)\in M} c_{ij}, \tag{12}$$

where F^* is the set of all matchings in G which differ from a maximum matching by at most $n\sqrt{n/m}$ edges. This problem can be solved in $O(n^{1.5}\sqrt{m})$ time by combining the maximum matching algorithm of Alt et al. [5] (see Section 1) with binary search. Then this solution is extended to an optimal solution of the bottleneck assignment

problem by growing at most $n\sqrt{n/m}$ augmenting paths. Every augmenting path can be completed in $O(m)$ time, see e.g. [147]. Thus the overall complexity of the algorithm becomes $O(n\sqrt{nm})$.

Pferschy [125] describes an algorithm with expected running time $O(n^2)$. Thus this algorithm is linear in m in the case that the graph G is dense. Pferschy's algorithm uses again the idea of thinning out the original problem by considering only the $2n \log n$ cheapest edges. This can be done in $O(n^2)$ time by using a linear selection algorithm for finding the $2n \log n$-smallest edge. In the second step the LBAP on the sparse subgraph is solved by using the method of Gabow and Tarjan. This yields $O((n \log n)^{3/2})$ additional elementary operations. Finally the solution is completed to a perfect matching in the full graph by applying again the algorithm of Gabow and Tarjan. But this completion step is only necessary with a low probability, namely with a probability less than $O(1/\sqrt{n \log n})$. Thus, the expected time needed by the completion is $O(n^2)$, and we get an overall expected running time of $O(n^2)$. This algorithm provides not only a good bound in terms of complexity, but is also very efficient and simple to use.

The so-called *balanced assignment problems* which were introduced by Martello et al. [110] are related to bottleneck assignment problems. Given a real $n \times n$ matrix $C = (c_{ij})$, the balanced assignment problem can be formulated as

$$\min_{\varphi} \left[\max_i c_{i\varphi(i)} - \min_i c_{i\varphi(i)} \right].$$

For solving this problem the authors sort the entries c_{ij} non-decreasingly and propose the following $O(n^4)$ procedure:

4.3.1. Algorithm for balanced assignment problems

1. Solve the corresponding bottleneck assignment problem. Let φ^* be its optimal solution.
2. Define
$$l := \min_i c_{i\varphi^*(i)}, \quad u := \max_i c_{i\varphi^*(i)}.$$
If $l = u$, stop. A balanced solution has been found. Otherwise go to Step 3.
3. Delete in C all elements $\leqslant l$ and $> u$ and grow augmenting paths.
If there exists a perfect matching in the corresponding graph, set φ^* equal to that solution and go to Step 2.
If no solution exists, then go to Step 4.
4. If u is already the maximum element of matrix C, stop. The present solution is optimal. Otherwise increase u to the next larger element and return to Step 3.

These balanced linear assignment problems can be used in a heuristic for decomposing traffic matrices arising from TDMA-systems (see Section 3) in at most n switch modes (cf. [8], who used LBAPs instead of balanced linear assignment problems in this context). Given the traffic matrix T, let φ^* be an optimal solution of the balanced assignment problem with coefficient matrix T. We set $\lambda_1 := \max_{1 \leqslant i \leqslant n} t_{i\varphi^*(i)}$ and forbid the elements $t_{i\varphi^*(i)}$, $i = 1, 2, \ldots, n$. With this new matrix we solve the next balanced assignment problem and determine λ_2. We continue in this way until all elements of T are forbidden. The rational behind this approach is that during the application of a fixed switch mode all involved stations should have about the same workload.

5. Multidimensional assignment problems

5.1. General multidimensional assignment problems

Multi-dimensional (sometimes called *multi-index*) assignment problems (MAP) have been introduced by Pierskalla [127] as natural extensions of linear assignment problems. The most prominent representatives of this class are axial and planar 3-dimensional assignment problems, which are treated in the subsections below. An annotated bibliography on this subject can be found in [22].

The axial MAP can be written as

$$
\begin{aligned}
\min \quad & \sum_{i_1=1}^{n} \cdots \sum_{i_d=1}^{n} c_{i_1 \cdots i_d} x_{i_1 \cdots i_d} \\
\text{s.t.} \quad & \sum_{i_2=1}^{n} \cdots \sum_{i_d=1}^{n} x_{i_1 \cdots i_d} = 1, \quad i_1 = 1, \ldots, n, \\
& \sum_{i_1=1}^{n} \cdots \sum_{i_{k-1}=1}^{n} \sum_{i_{k+1}=1}^{n} \cdots \sum_{i_d=1}^{n} x_{i_1 \cdots i_d} = 1, \\
& \qquad \text{for } k = 2, \ldots, d-1, \quad \text{and} \quad i_k = 1, 2, \ldots, n, \\
& \sum_{i_1=1}^{n} \cdots \sum_{i_{d-1}=1}^{n} x_{i_1 \cdots i_d} = 1, \quad i_d = 1, \ldots, n, \\
& x_{i_1 \cdots i_d} \in \{0, 1\} \quad \text{for } 1 \leqslant i_1, i_2, \ldots, i_d \leqslant n,
\end{aligned} \tag{13}
$$

with n^d cost coefficients $c_{i_1 \cdots i_d}$.

More simply, we just asks for $d-1$ permutations $\varphi_1, \varphi_2, \ldots, \varphi_{d-1}$ which minimize the objective function

$$\sum_{i=1}^{n} c_{i\varphi_1(i)\varphi_2(i) \ldots \varphi_{d-1}(i)}.$$

The multidimensional assignment problem is NP-hard in general, but in the case that the array of the cost coefficients is a Monge array (see [38]), it is solved by the identical permutations $\varphi_i = id$, for $i = 1, 2, \ldots, d-1$. Burkard et al. [42] have shown, however, that the MAP remains NP-hard for $d \geqslant 3$, if the cost array fulfills an Anti Monge condition. (Remember that for $n = 2$ an LSAP with an Anti-Monge cost matrix is solved by the permutation φ with $\varphi(i) = n - i + 1$.) This implies that *maximizing* the objective function with cost elements drawn from a Monge array is NP-hard.

General multidimensional assignment problems have recently been considered to model data association problems in connection with multi-target tracking and multi-sensor surveillance, see Poore [128] and Poore et al. [131] for more details. These authors solve the occurring MAPs by Lagrangean relaxation methods, see [130,132]. A numerical study of data association problems arising in multi-target and multi-sensor tracking is given in Poore [129]. Greedy randomized adaptive search (GRASP)

heuristics for multidimensional assignment problems arising in multitarget tracking and data association have been proposed by Murphey et al. [116,117].

Pusztaszeri et al. [134] describe another interesting MAP which arises in the context of tracking elementary particles. By solving a five-dimensional assignment problem, they reconstruct tracks of charged elementary particles generated by the Large Electron-Positron Collider at CERN in Geneva.

5.2. Axial three-dimensional assignment problems

Consider n^3 cost coefficients c_{ijk}. The three-dimensional axial assignment problem (3-DAP) can be described with the help of two permutations φ and ψ as

$$\min_{\varphi,\psi\in\mathscr{S}_n} \sum_{i=1}^{n} c_{i\varphi(i)\psi(i)}. \tag{14}$$

Its name stems from the formulation

$$\begin{aligned}
\min \quad & \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} c_{ijk}x_{ijk} \\
\text{s.t.} \quad & \sum_{j=1}^{n}\sum_{k=1}^{n} x_{ijk} = 1, \quad i = 1,2,\ldots,n, \\
& \sum_{i=1}^{n}\sum_{k=1}^{n} x_{ijk} = 1, \quad j = 1,2,\ldots,n, \\
& \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ijk} = 1, \quad k = 1,2,\ldots,n, \\
& x_{ijk} \in \{0,1\} \quad \text{for all } 1 \leqslant i,j,k \leqslant n,
\end{aligned} \tag{15}$$

where the 1-s on the right-hand side are assigned to positions at the axes of a 3-dimensional array. The sum over the corresponding "flat" in the array must equal the amount assigned to the position on the axis, see Fig. 5.

According to (14) 3-DAP has $(n!)^2$ feasible solutions. Karp [93] showed that the 3-DAP is $\mathscr{NP}$-hard.

Euler [65] started the investigation of the axial 3-index assignment polytope, i.e., the convex hull of feasible solutions to problem (15). He considers the role of odd cycles for a class of facets of this polytope. Independently, Balas and Saltzman [10] investigate in detail the polyhedral structure of the three-index assignment polytope. They show that this polytope has dimension $n^3 - 3n + 2$ and they describe an $O(n^4)$ separation algorithm for facets induced by certain cliques. Balas and Qi [9] and Qi et al. [135] continue the above work.

Several branch and bound algorithms were proposed for solving 3-DAPs. Most of these algorithms split the current problem into two subproblems by fixing one variable x_{ijk} to 1 and to 0, respectively. Balas and Saltzman [11] introduced a branching strategy which exploits the structure of the problem and allows to fix several variables at each branching node. Hansen and Kaufman [84] describe a primal-dual method similar to

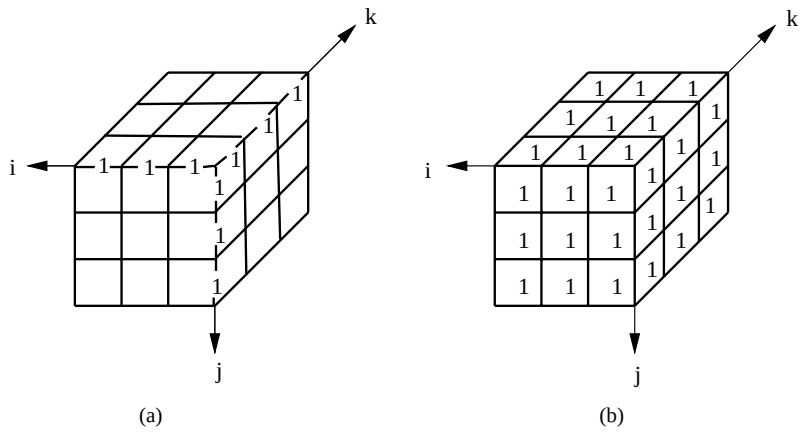

Fig. 5. (a) A geometric representation of the axial 3-dimensional assignment problem for $n = 3$. (b) A geometric representation of the planar 3-dimensional assignment problem for $n = 3$.

the Hungarian method for linear assignment problems. First as many 0-elements as possible are generated among the cost coefficients by generalized row reductions and column reductions. Then a covering problem is solved for these 0-elements. If the covering number is smaller than n, further 0-elements are generated by means of an admissible transformation similar to that in Section 4.1. If the covering number equals n, the corresponding stability problem in a hypergraph is solved. This stability problem replaces the determination of a maximum matching in the 2-dimensional case. The minimum number of (hyper-)edges in a cover is in general strictly larger than the cardinality of a stable set. If the stability number equals n, an optimal solution has been found, otherwise a branching is performed. Note the similarity of this method with the approach in Section 4.1 for solving algebraic assignment problems. Due to this similarity the same algorithm solves the "*algebraic*" version of the 3-DAP with cost coefficients drawn from a d-monoid. The subproblems of finding a minimum cover and a maximum stable set are in the 3-dimensional case, however, $\mathcal{NP}$-hard.

Due to Burkard and Fröhlich [35] admissible transformations for 3-DAPs have the following form. Let us first introduce some notation. Let $N := \{1, 2, \ldots, n\}$, $I, J, K \subseteq N$ and denote $\bar{I} := N \setminus I$, $\bar{J} = N \setminus J$, $\bar{K} = N \setminus K$. With these definitions we get

Theorem 5.1 (Admissible transformations for the 3-DAP [35]). *Let a 3-DAP with cost coefficients c_{ijk}, $i, j, k = 1, 2, \ldots, n$, be given. For $I, J, K \subseteq N$ with $m := n - (|I| + |J| + |K|) \geqslant 1$ and*

$$c := \min\{c_{ijk} : (i, j, k) \in \bar{I} \times \bar{J} \times \bar{K}\}$$

we define

$$\bar{c}_{ijk} := c_{ijk} - c, \quad (i, j, k) \in \bar{I} \times \bar{J} \times \bar{K}$$

$$\bar{c}_{ijk} := c_{ijk} + c, \quad (i, j, k) \in (\bar{I} \times J \times K) \cup (I \times \bar{J} \times K) \cup (I \times J \times \bar{K})$$

$$\bar{c}_{ijk} := c_{ijk} + 2c, \quad (i,j,k) \in I \times J \times K$$

$$\bar{c}_{ijk} := c_{ijk}, \quad \textit{otherwise.}$$

Then, for any feasible solution φ, ψ *of the* 3-*DAP we have*

$$\sum_{i=1}^{n} c_{i\varphi(i)\psi(i)} = \sum_{i=1}^{n} \bar{c}_{i\varphi(i)\psi(i)} + mc.$$

Row and column reductions are special cases of the admissible transformations described by the above theorem.

Strong lower bounds are essential for a branch and bound procedure. In the case of 3-DAP lower bounds can be computed by the following Lagrangean relaxation approach. Let us take two blocks of the constraints in (15) into the objective function via Lagrangean multipliers (cf. [35]):

$$L(\pi,\varepsilon) := \min \left\{ \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} (c_{ijk} + \pi_j + \varepsilon_i)x_{ijk} - \sum_{j=1}^{n} \pi_j - \sum_{i=1}^{n} \varepsilon_i \right\}$$

such that

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ijk} = 1, \quad k = 1,2,\ldots,n$$

$$x_{ijk} \in \{0,1\}, \quad 1 \leqslant i,j,k \leqslant n$$

$$\pi \in \mathbb{R}^n, \ \varepsilon \in \mathbb{R}^n.$$

Since $L(\pi,\varepsilon)$ is a concave function, we can use a subgradient method for finding its maximum.

5.2.1. Algorithm for maximizing $L(\pi,\varepsilon)$

1. Start with $r = 0$, $\pi^r := \varepsilon^r := 0$.
2. Use a greedy algorithm to minimize $L(\pi^r, \varepsilon^r)$. Let x^r_{ijk} be the corresponding optimal solution.
3. Define

$$v^r_{i_0} := |\{x^r_{i_0,j,k} : x_{i_0,j,k} = 1\}| - 1 \quad \text{for } i_0 = 1,2,\ldots,n$$

and

$$w^r_{j_0} := |\{x^r_{i,j_0,k} : x_{i,j_0,k} = 1\}| - 1 \quad \text{for } j_0 = 1,2,\ldots,n.$$

4. If $v^r = w^r = (0,0,\ldots,0)$, then the maximum is reached. Terminate.
5. If a prespecified number of iterations is not yet reached, update π and ε by setting

$$\pi^{r+1} := \pi^r + \lambda_r w^r,$$

$$\varepsilon^{r+1} := \varepsilon^r + \lambda_r v^r,$$

where λ_r is a suitable step length. Go to Step 2.
Otherwise terminate.

Another subgradient procedure for solving a Lagrangean relaxation of the 3-DAP together with computational considerations has been described by Frieze and Yadegar [72]. Burkard and Rudolf [41] report on satisfactory computational results obtained by an algorithm which uses the classical branching rule combined with a reduction step in every node of the search tree. The lower bound computation is done by applying the above described subgradient optimization procedure.

There exists a number of polynomially solvable special cases of the 3-DAP. As mentioned in Section 5.1, the 3-DAP becomes polynomially solvable, if the cost coefficients are taken from a 3-dimensional Monge array (see [38]). Burkard et al. [42] investigate 3-DAPs with decomposable cost coefficients, where $c_{ijk} = u_i v_j w_k$ and u_i, v_j, and w_k are non-negative. They show that the maximization version of this problem is polynomially solvable, whereas the minimization is in general $\mathcal{NP}$-hard. Moreover, several polynomially solvable special cases of the minimization problem are identified.

5.3. Planar three-dimensional assignment problems

Planar 3-*dimensional assignment problems* (3-PAP) have the form:

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} c_{ijk}x_{ijk}$$

s.t.

$$\sum_{i=1}^{n} x_{ijk} = 1, \quad j,k = 1,2,\ldots,n,$$

$$\sum_{j=1}^{n} x_{ijk} = 1, \quad i,k = 1,2,\ldots,n,$$

$$\sum_{k=1}^{n} x_{ijk} = 1, \quad i,j = 1,2,\ldots,n,$$

$$x_{ijk} \in \{0,1\}, \quad i,j,k = 1,2,\ldots,n. \tag{16}$$

3-PAPs play a crucial role in the context of time tabling problems. For a geometric interpretation of planar 3-dimensional assignment problems see Fig. 5. Every "flat" in the three-dimensional array x_{ijk} must contain a (2-dimensional) assignment. Thus the feasible solutions of the 3-PAP correspond to *Latin squares*. Fig. 6 shows a feasible solution for a 3-PAP with $n = 3$: number 1 represents the assignment in the lowest horizontal flat, number 2 shows the assignment in the medium flat, and 3 represents the assignment in the upper flat. Due to this interpretation, the number of feasible solutions of a 3-PAP of size n equals the number of Latin squares of order n, and hence increases very fast. Due to Bammel and Rothstein [12] the number of feasible solutions for a 3-PAP with $n = 9$ is $9! \cdot 8! \cdot 377,597,570,964,258,816$.

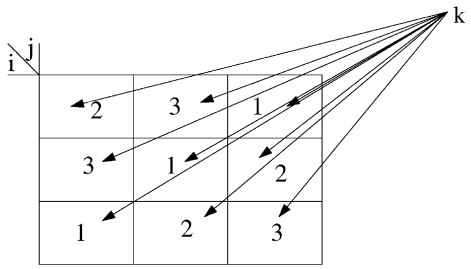

Fig. 6. A latin square representing a feasible solution of the planar 3-dimensional assignment problem of size $n = 3$.

Frieze [71] showed that the 3-PAP is $\mathcal{NP}$-hard. A partial description of the polyhedral structure of the 3-PAP polytope can be found in Euler et al. [66]. See also the related study on time tabling polyhedra by Euler and Verge [67].

There are not many algorithms known for the 3-PAP. The first branch and bound algorithm goes back to Vlach [150]. Vlach computes lower bounds by applying (generalized) row and column reductions similar to those used in the case of 3-DAPs. Another branch and bound procedure for the planar 3-index assignment problem has been described by Magos and Miliotis [109] who also report computational results. Later, Magos [108] used similar ideas for implementing a tabu search algorithm for 3-PAPs. A move in the neighborhood of some Latin square is completely determined by changing the contents of a certain cell (i, j). This affects at least 4 and at most $2n$ other cells which have to be adapted accordingly. Since there are n^2 cells in total and each cell may take $n-1$ new values, the neighborhood size lies between $n(n-1)/2$ and $n^2(n-1)/4$. This neighbourhood structure has two nice properties: first, the change in the objective function value after each move can be computed in linear time. Secondly, not all moves have to be evaluated in each iteration: all moves which put a certain element in a certain cell imply the same change in the objective function, independently from the solution to which they are applied. The numerical results with this algorithm show a good trade-off between computation time and solution quality for 3-PAP instances of size up to $n = 14$.

6. Quadratic assignment problems

6.1. Problem statement, applications and complexity

The quadratic assignment problem (QAP) was introduced by Koopmans and Beckmann [100] in 1957 as a mathematical model for the location of indivisible economical activities. For a comprehensive survey of this field see [28], for an annotated bibliography contact [22]. First we give a description of a QAP as a locational problem. Let us assign n facilities to n locations with the cost being proportional to the flow between the facilities multiplied with their distances plus costs associated with a facility being

placed at a certain location. The objective is to allocate each facility at a location such that the total cost is minimized. Thus we are given three $n \times n$ matrices, the flow matrix $A = (a_{ij})$, the distance matrix $B = (b_{kl})$ and matrix $C = (c_{ik})$, where c_{ik} is the cost of placing facility i at location k. The QAP in Koopmans–Beckmann form can now be written as

$$\min_{\varphi \in \mathscr{S}_n} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\varphi(i)\varphi(j)} + \sum_{i=1}^{n} c_{i\varphi(i)}. \tag{17}$$

Each individual product $a_{ij} b_{\varphi(i)\varphi(j)}$ is the cost caused by assigning facility i to location $\varphi(i)$ and facility j to location $\varphi(j)$. An instance of the QAP with input matrices A, B and C will be denoted by $QAP(A, B, C)$.

A more general version of the QAP was introduced by Lawler [103]. Lawler considers a four-dimensional cost array $C = (c_{ijkl})$ instead of the two matrices A and B, where the linear term C can be added to the elements c_{ijij}. Thus we get the general form of a QAP as

$$\min_{\varphi \in \mathscr{S}_n} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij\varphi(i)\varphi(j)}. \tag{18}$$

It is astonishing how many real life applications can be modeled as QAPs. A natural application in location theory was used by Dickey and Hopkins [60] in a *campus planning* model. The problem consists of planning the sites of n buildings on a campus, where b_{kl} is the distance from site k to site l, and a_{ij} is the traffic intensity between building i and building j. The objective is to minimize the total weekly walking distance between the buildings.

In addition to facility location QAPs appear in applications such as layout problems, backboard wiring, computer manufacturing, scheduling, process communications and turbine balancing (see Section 6.5). In the field of ergonomics Burkard and Offermann [39] showed that QAPs can be applied to *typewriter keyboard design*. The problem is to arrange the keys on a keyboard such as to minimize the time needed to write some text. Let the set of integers $N = \{1, 2, \ldots, n\}$ denote the set of symbols to be arranged. Then a_{ij} denotes the frequency of the appearance of the pair of symbols i and j. The entries of the distance matrix b_{kl} are the times needed to press the key in position l after pressing the key in position k. A permutation $\varphi \in \mathscr{S}_n$ describes an assignment of symbols to keys. An optimal solution φ^* for the QAP minimizes the average time for writing a text. A similar application related to an ergonomic design is the development of control boards in order to minimize eye fatigue by McCormick [112]. Further applications concern the ranking of archeological data [101], the ranking of a team in a relay race [86], scheduling parallel production lines [77], and analyzing chemical reactions for organic compounds [149].

By replacing the sums in the objective function of a QAP by the maximum operation, we get the so-called *quadratic bottleneck assignment problem* (BQAP). A BQAP (in Koopmans–Beckmann form) can be formulated as

$$\min_{\varphi} \max_{1 \leqslant i,j \leqslant n} a_{ij} b_{\varphi(i)\varphi(j)}.$$

The first occurrence of the BQAP is due to Steinberg [145] and arises as an application in backboard wiring while trying to minimize the maximum length of the involved wires. Another important application of the BQAP, the *bandwidth minimization problem* stems from numerical analysis. In the bandwidth problem we want to find a permutation of the rows and columns of a given matrix such that after permuting the rows and columns the new matrix has minimum bandwidth. It is easy to see that this problem can be modeled as a special BQAP with a 0–1 flow matrix which has an 1-entry, iff the given matrix has at this position a non-zero entry. Defining the distance matrix $B = (b_{kl})$ by $b_{kl} := |k - l|$ and solving the corresponding BQAP leads to an optimal solution of the bandwidth minimization problem.

Besides these applications basically all QAP applications give rise to a BQAP model as well, because it often makes sense to minimize the largest cost instead of the overall cost incurred by some decision.

In contrast to linear assignment problems, quadratic assignment problems remain among the hardest combinatorial optimization problems. Marzetta and Brüngger [111] report recently on the solution of a Koopmans Beckmann problem of size $n = 25$ of the QAP-library (see Section 6.4 and Burkard et al. [37]). The inherent difficulty for solving QAPs is also reflected by their computational complexity. Sahni and Gonzalez [141] showed that the QAP is $\mathcal{NP}$-hard and that even finding an approximate solution within some constant factor from the optimum value cannot be done in polynomial time unless $\mathcal{P} = \mathcal{NP}$. These results hold even for Koopmans–Beckmann QAPs with coefficient matrices fulfilling the triangle inequality, see Queyranne [136]. The *linear dense arrangement problem*, however, which is a special Koopmans–Beckmann QAP, admits a *polynomial time approximation scheme* ($PTAS$), see Arora et al. [6]. In the linear dense arrangement problem matrix A is the distance matrix of n points which are regularly spaced on a line, i.e., points with abscissae given by $x_p = p,\ p = 1,\ldots,n$ and B is a dense 0–1 matrix, i.e., the number of 1-entries in B is in $\Omega(n^2)$.

Recently it has been shown that even local search is hard in the case of the QAP. It can be shown (see [124] for details) that the QAP with respect to the pairwise exchange neighborhood structure as well as with respect to a Lin–Kernighan-like neighbourhood structure [118] is $\mathcal{PLS}$-complete. This implies that the time complexity of a local search method for the QAP using either of the two mentioned neighbourhood structures is exponential in the worst case. Moreover, from results of Papadimitriou and Wolfe [123] follows that deciding whether a given local optimal solution of the QAP is also globally optimum, is $\mathcal{NP}$-complete.

The quadratic bottleneck assignment problem is $\mathcal{NP}$-hard as well, since it contains the bottleneck travelling salesman problem as special case and therefore the problem to decide whether a given graph contains a Hamiltonian cycle or not. To see this, consider a $QAP(A, B)$ where A is the adjacency matrix of the given graph and B is the permutation matrix of a cyclic permutation.

6.2. Different problem formulations and linearizations

There exist different, but equivalent mathematical formulations for QAPs which stress different structural characteristics of the problem and lead to different solution

approaches. It is immediate that we can write (17) as an integer quadratic program of the form

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} a_{ij}b_{kl}x_{ik}x_{jl} + \sum_{i,j=1}^{n} c_{ij}x_{ij} \tag{19}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_{ij} = 1, \quad j = 1,2,\ldots,n, \tag{20}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1,2,\ldots,n, \tag{21}$$

$$x_{ij} \in \{0,1\}, \quad i,j = 1,2,\ldots,n. \tag{22}$$

We can formulate this Koopmans–Beckmann QAP in a more compact way by defining an *inner product* between matrices. Let the inner product of two real $n \times n$ matrices A, B be defined by

$$\langle A, B\rangle := \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}b_{ij}.$$

Given some $n \times n$ matrix B and a permutation $\varphi \in \mathscr{S}_n$ with the associated permutation matrix $X_\varphi \in \mathbf{X}_n$, we have

$$X_\varphi B X_\varphi^{\mathrm{T}} = (b_{\varphi(i)\varphi(j)}). \tag{23}$$

Thus a Koopmans–Beckmann QAP can be written as

$$\begin{aligned} \min \quad & \langle A, XBX^{\mathrm{T}}\rangle + \langle C, X\rangle \\ \text{s.t.} \quad & X \in \mathbf{X}_n. \end{aligned} \tag{24}$$

Formulation (17) together with (23) lead immediately to the so-called *trace formulation* of a Koopmans–Beckmann problem. Recall that the trace on an $n \times n$ matrix is defined as sum of its diagonal elements. Therefore

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}b_{\varphi(i)\varphi(j)} = tr(A\bar{B})$$

with $\bar{B} = XB^{\mathrm{T}}X^{\mathrm{T}}$. Since $tr(CX^{\mathrm{T}}) = \sum_{i=1}^{n} c_{i\varphi(i)}$, the QAP in (24) can be formulated as

$$\begin{aligned} \min \quad & tr(AXB^{\mathrm{T}} + C)X^{\mathrm{T}} \\ \text{s.t.} \quad & X \in \mathbf{X}_n. \end{aligned} \tag{25}$$

The trace formulation of the QAP appeared first in Edwards [64], and was used by Finke et al. [68] to introduce eigenvalue bounds for QAPs (see Section 6.3.2).

In the general case, let the coefficients c_{ijkl} be the entries of an $n^2 \times n^2$ matrix S such that c_{ijkl} lies in row $(i-1)n+k$ and column $(j-1)n+l$. Since $x^{\mathrm{T}}[1/2(S+S^{\mathrm{T}})]x = x^{\mathrm{T}}Sx$, we can assume that S is symmetric. The addition of a constant to the entries of the main diagonal of S does not change the optimal solutions of the corresponding QAP,

it simply adds a constant to the objective function value. Thus we can assume that S is positive definite or we can also assume that S is negative definite. Let $x = (x_{11}, x_{12}, \ldots, x_{1n}, x_{21}, \ldots, x_{nn})^{\mathrm{T}} = (x_1, \ldots, x_{nn})^{\mathrm{T}}$. Then we can write a QAP as quadratic convex program (quadratic concave program) in the form

$$\begin{aligned}
\min \quad & x^{\mathrm{T}} S x \\
\text{s.t.} \quad & \sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, 2, \ldots, n, \\
& \sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, 2, \ldots, n, \\
& x_{ij} \geqslant 0, \quad i, j = 1, 2, \ldots, n. \qquad (26)
\end{aligned}$$

where S is symmetric and positive (negative) definite.

Many authors have proposed methods for linearizing the quadratic form in the objective function (19) by introducing additional variables. Lawler [103] replaces the quadratic terms $x_{ij}x_{kl}$ in the objective function of (18) by n^4 variables

$$y_{ijkl} := x_{ij}x_{kl}, \quad i, j, k, l = 1, 2, \ldots, n, \qquad (27)$$

and obtains in this way a 0–1 linear program with $n^4 + n^2$ binary variables and $n^4 + 2n + 1$ constraints. The QAP (19)–(22) can be written as a 0–1 linear program in the following form (see [103,20])

$$\begin{aligned}
\min \quad & \sum_{i,j=1}^{n} \sum_{k,l=1}^{n} c_{ijkl} y_{ijkl} \\
\text{s.t.} \quad & \sum_{i,j=1}^{n} \sum_{k,l=1}^{n} y_{ijkl} = n^2, \qquad (28) \\
& x_{ij} + x_{kl} - 2y_{ijkl} \geqslant 0, \quad i, j, k, l = 1, 2, \ldots, n, \\
& y_{ijkl} \in \{0, 1\}, \quad i, j, k, l = 1, 2, \ldots, n, \\
& (x_{ij}) \in \mathbf{X}_n. \qquad (29)
\end{aligned}$$

Kaufman and Broeckx [97] introduced a linearization which yields the smallest number of additionally introduced variables and constraints. Their model employs n^2 real variables, n^2 binary variables and $n^2 + 2n$ constraints. Frieze and Yadegar [73] get a mixed integer linear programming formulation for the QAP with n^4 real variables, n^2 binary variables and $n^4 + 4n^3 + n^2 + 2n$ constraints.

Based on a linearization technique for general 0–1 polynomial programs due to Adams and Sherali [2,3], Adams and Johnson [1] present an 0–1 linear integer programming formulation for the QAP, which resembles to a certain extent the linearization of Frieze and Yadegar. They show that a QAP can be written as the following

mixed 0–1 linear program

$$\min \quad \sum_{i,j=1}^{n} \sum_{k,l=1}^{n} c_{ijkl}\, y_{ijkl}$$

s.t.

$$\sum_{i=1}^{n} y_{ijkl} = x_{kl}, \quad j,k,l = 1,2,\ldots,n,$$

$$\sum_{j=1}^{n} y_{ijkl} = x_{kl}, \quad i,k,l = 1,2,\ldots,n,$$

$$y_{ijkl} = y_{klij}, \quad i,j,k,l = 1,2,\ldots,n,$$

$$y_{ijkl} \geqslant 0, \quad i,j,k,l = 1,2,\ldots,n,$$

$$(x_{ij}) \in \mathbf{X}_n, \tag{30}$$

where each y_{ijkl} represents the product $x_{ij}x_{kl}$. The above formulation contains n^2 binary variables x_{ij}, n^4 continuous variables y_{ijkl} and $n^4 + 2n^3 + 2n$ constraints excluding the non-negativity constraints on the continuous variables. Although a significant smaller formulation in terms of the number of variables and constraints could be obtained, the structure of the above formulation is favorable for solving QAPs approximately by means of the Lagrangean dual.

Closely related to linearizations are polyhedral studies of the QAP which have recently be performed by Barvinok [13], Jünger and Kaibel [89,90], Kaibel [91], and Padberg and Rijal [121]. The *QAP polytope* can be defined by using the following graph theoretical model:

For each $n \in \mathbb{N}$ consider a graph $G_n=(V_n, E_n)$ with vertex set $V_n=\{(i,j)\colon 1 \leqslant i,j \leqslant n\}$ and edge set $E_n = \{((i,j),(k,l))\colon i \neq k, j \neq l\}$. The maximum cliques in G_n have cardinality n and correspond to permutations. Given an instance of the QAP with coefficients c_{ijkl} we introduce the coefficients c_{ijij} as vertex weights and c_{ijkl} with $i \neq k$, $j \neq l$ as weight of the edge $((i,j),(k,l))$. Solving the QAP is equivalent to finding a maximal clique with minimum total vertex weight and edge weight. For each maximum clique C in G_n we denote its incidence vector by (x^C, y^C), where

$$x_{ij}^C = \begin{cases} 1 \text{ if } (i,j) \text{ is a vertex of the clique } C, \\ 0 \text{ otherwise} \end{cases}$$

and

$$y_{ijkl}^C = \begin{cases} 1 \text{ if } (i,j),(k,l) \text{ is an edge of clique } C, \\ 0 \text{ otherwise.} \end{cases}$$

The QAP polytope QAP_n is then defined as convex hull of all vectors (x^C, y^C), where C is a maximum clique in G_n. It turns out that *the traveling salesman polytope* and *the linear ordering polytope* are projections of QAP_n, and that QAP_n is a face of the *Boolean quadric polytope*, see [91].

Barvinok [13], Padberg and Rijal [121], as well as Jünger and Kaibel [89] compute independently the dimension of QAP_n and show that the inequalities $y_{ijkl} \geqslant 0$, $i \neq k$, $j \neq l$, are facet defining. These facets are usually called the *trivial facets* of QAP_n. Moreover, Padberg and Rijal [121], and Jünger and Kaibel [89] show that the affine hull of QAP_n is described by the following equations which are linearly independent:

$$\sum_{i=1}^{n} x_{ij} = 1, \quad 1 \leqslant j \leqslant n-1, \tag{31}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad 1 \leqslant i \leqslant n, \tag{32}$$

$$-x_{kl} + \sum_{i=1}^{k-1} y_{ijkl} + \sum_{i=k+1}^{n} y_{klij} = 0 \quad \begin{array}{l} 1 \leqslant j \neq l \leqslant n, 1 \leqslant k \leqslant n-1, \\ \text{or} \quad 1 \leqslant l < j \leqslant n, \ k = n, \end{array} \tag{33}$$

$$-x_{ij} + \sum_{l=1}^{j-1} y_{ijkl} + \sum_{l=j+1}^{n} y_{ijkl} = 0 \quad \begin{array}{l} 1 \leqslant j \leqslant n, 1 \leqslant i \leqslant n-3, \\ i < k \leqslant n-1 \quad \text{or} \\ 1 \leqslant j \leqslant n-1, \ i = n-2, \\ k = n-1, \end{array} \tag{34}$$

$$-x_{kj} + \sum_{l=1}^{j-1} y_{ilkj} + \sum_{l=j+1}^{n} y_{ilkj} = 0 \quad \begin{array}{l} 1 \leqslant j \leqslant n-1, 1 \leqslant i \leqslant n-3, \\ i < k \leqslant n-1. \end{array} \tag{35}$$

Summarizing we get the following theorem:

Theorem 6.1.

(i) *The affine hull of the QAP polytope QAP_n is given by the linear equations* (31)–(35). *These equations are linearly independent. The rank of the system is* $2n(n-1)^2 - (n-1)(n-2)$, *for* $n \geqslant 3$.

(ii) *For* $n \geqslant 3$ *the dimension of QAP_n is equal to* $1+(n-1)^2+n(n-1)(n-2)(n-3)/2$.

(iii) *The inequalities* $y_{ijkl} \geqslant 0$, $i < k$, $j \neq l$, *define facets of QAP_n.*

Padberg and Rijal [121] identify additionally two classes of valid inequalities for QAP_n, the *clique* inequalities and the *cut* inequalities, where the terminology is related to the graph G_n. The authors specify some conditions under which the cut inequalities are not facet defining. It is an open problem, however, to identify facet defining inequalities within these classes. Further valid inequalities, the so-called *box inequalities* have been described by Kaibel [91]. These inequalities are obtained by exploiting the relationship between the Boolean quadric polytope and the QAP polytope. For box inequalities it can be decided in polynomial time whether they are facet defining or not, and in the latter case some dominating facet defining inequality can be derived.

Similar results have been obtained for the *symmetric QAP polytope $SQAP_n$* arising in the case that at least one of the coefficient matrices A or B in a Koopmans–Beckmann problem is symmetric. $SQAP_n$ is defined by means of a hypergraph $H_n = (V_n, F_n)$,

where V_n is the same set of vertices as in graph G_n and F_n is the set of hyperedges $\{(i,j),(k,l),(i,l),(k,j)\}$ for all $i \neq k,\ j \neq l$. A set $C \subset V_n$ is called a clique in H_n if it is a clique in G_n. Again, the incidence vector (x^C, y^C) of a clique C is introduced by,

$$x_{ij} = \begin{cases} 1, \text{ if } (i,j) \text{ is a vertex of clique } C \\ 0, \text{ otherwise} \end{cases}$$

and

$$y_{ijkl} = \begin{cases} 1, \text{ if } i < k,\ l \neq j,\ \{(i,j),(k,l),(i,l),(k,j)\} \text{ is a hyperedge of clique } C \\ 0, \text{ otherwise.} \end{cases}$$

The convex hull of all incidence vectors (x^C, y^C) is called the symmetric QAP polytope $SQAP_n$.

Padberg and Rijal [121] and Jünger and Kaibel [90] give the following minimal description for the affine hull of $SQAP_n$:

$$\sum_{j=1}^{n} x_{ij} = 1, \quad 1 \leqslant i \leqslant n, \tag{36}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad 1 \leqslant j \leqslant n-1, \tag{37}$$

$$-x_{ij} - x_{kj} + \sum_{l=1}^{j-1} y_{ilkj} + \sum_{l=j+1}^{n} y_{ijkl} = 0 \quad \begin{array}{l} 1 \leqslant i < k \leqslant n, \\ 1 \leqslant j \leqslant n, \end{array} \tag{38}$$

$$-x_{kj} - x_{kl} + \sum_{i=1}^{k-1} y_{ijkl} + \sum_{i=k+1}^{n} y_{kjil} = 0 \quad \begin{array}{l} 1 \leqslant k \leqslant n, \\ 1 \leqslant j \leqslant n-3, \\ 1 \leqslant j < l \leqslant n-1. \end{array} \tag{39}$$

The results concerning $SQAP_n$ can be summarized in the following theorem:

Theorem 6.2.

(i) *The affine hull of the symmetric QAP polytope $SQAP_n$ is described by the linear equations* (36)–(39). *These equations are linearly independent, their rank is* $n^2(n-2)+2n-1$.

(ii) *The dimension of $SQAP_n$ is equal to* $(n-1)^2 + n^2(n-3)^2/4$.

(iii) *The inequalities* $y_{ijkl} \geqslant 0$ *for* $i < k,\ j < l$, *and* $x_{ij} \geqslant 0$ *for* $1 \leqslant i,j \leqslant n$, *define facets of $SQAP_n$.*

(iv) *For each* $i < k$ *and for all* $J \subseteq \{1,2,\ldots,n\}$ *the* row curtain inequalities

$$-\sum_{j \in J} x_{ij} + \sum_{\substack{j,l \in J \\ j<l}} y_{ijkl} \leqslant 0$$

are valid for $SQAP_n$. For each $j < l$ and for all $I \subseteq \{1,2,\ldots,n\}$ the column curtain inequalities

$$-\sum_{i\in I} x_{ij} + \sum_{\substack{i,k\in I\\ i<k}} y_{ijkl} \leqslant 0$$

are valid for $SQAP_n$.

All curtain inequalities with $3 \leqslant |I|, |J| \leqslant n-3$ define facets of $SQAP_n$. The other curtain inequalities define faces which are contained in trivial facets of $SQAP_n$.

(v) *The separation problem for curtain inequalities is $\mathcal{NP}$-hard.*

6.3. Lower bounds

Since QAPs are $\mathcal{NP}$-hard, good lower bounds are of eminent importance for solving these problems by implicit enumeration procedures like branch and bound. We require for a good bound that it is not too hard to compute, that it can easily be evaluated for subsets of the problem which occur after some branching and, finally, that it is tight. There are many different proposals for deriving bounds. In the following we survey briefly bounds based on linearizations of the QAP and eigenvalue bounds which are related to the trace formulation of the QAP.

6.3.1. Bounds based on linearizations

Let us consider a Koopmans–Beckmann problem QAP(A,B,C). W.l.o.g. we can assume that all entries in the matrices A and B are non-negative. For each row index i let $\hat{a}_{(i,.)}$ be the $(n-1)$-dimensional vector obtained from the ith row of A by deleting the element a_{ii}. Similarly define $\hat{b}_{(k,.)}$ for every row k of matrix B. According to Theorem 4.5 we get the minimum scalar product $\langle a,b\rangle^- := \min_\varphi \sum_{i=1}^n a_i b_{\varphi(i)}$ of two non-negative vectors $a,b\in\mathbb{R}^n$ by sorting the elements of a non-decreasingly and the elements of b non-increasingly. Thus $\langle \hat{a}_{(i,.)}, \hat{b}_{(k,.)}\rangle^-$ is the minimum cost which occurs if index i is mapped to index k. In order to find a lower bound on the value of a QAP we first compute the n^2 minimum scalar products $\langle \hat{a}_{(i,.)}, \hat{b}_{(k,.)}\rangle^-$ and define a new cost matrix $L = (l_{ik})$ by

$$l_{ik} = a_{ii}b_{kk} + c_{ik} + \langle \hat{a}_{(i,.)}, \hat{b}_{(k,.)}\rangle^-. \tag{40}$$

We obtain the Gilmore–Lawler lower bound GLB for the Koopmans–Beckmann QAP by solving the linear assignment problem with cost matrix L. The appropriate sorting of the rows and columns of A and B can be done in $O(n^2 \log n)$ time. The computation of all l_{ik} takes $O(n^3)$ time and the same amount of time is needed to solve the last LAP. Thus the Gilmore–Lawler bound for Koopmans–Beckmann problems can be computed in $O(n^3)$ time. Thus it is easy to compute, but it deteriorates fast as the size of the problems increases. A very similar procedure is possible for general QAPs, see e.g. the handbook article of Burkard and Çela [24].

The Gilmore–Lawler bound can be strengthened by splitting the coefficients a_{ij} and b_{kl} and thus transfering some amount from the quadratic part of the objective function to

the linear part of the objective function. This can be done by defining new coefficients $\bar{a}_{ij}, \bar{b}_{kl}, \lambda_i$ and μ_k by the formulas

$$a_{ij} = \bar{a}_{ij} + \lambda_i,$$

$$b_{kl} = \bar{b}_{kl} + \mu_k,$$

where the amounts λ_i and μ_k are suitably chosen, e.g. as row minima. Such a *reduction* was first used by Conrad [52] and later independently investigated by many researchers (see [17,140,64,73]). Similar procedures can also be applied to the bottleneck QAP, see [18].

There are several bounding strategies which are closely related to the Gilmore–Lawler bound and reductions. One of them is the bounding strategy of Hahn and Grant [82]. This procedure combines GLB ideas with reduction steps in a dual framework. Other possibilities are exploited by reformulation methods in which the coefficients of the problem are changed such that the new problem has the same objective function value as the original problem for any permutation matrix $X = (x_{ij}) \in \mathbf{X}_n$, but a stronger bound can be derived. Reformulation rules stem from Carraresi and Malucelli [46] and from Assad and Xu [7].

A further way to strengthen GLB stems from Frieze and Yadegar [73]. These authors start from their linearization and include some of the constraints via Lagrangean multipliers in the objective function. The corresponding Lagrangean problem can be solved by subgradient methods and yields sharper bounds than GLB. A similarly approach is used by Adams and Johnson [1]. They add the so-called *complementary constraints*

$$y_{ijkl} = y_{klij}$$

to the objective function via Lagrangean multipliers α_{ijkl} and obtain a Lagrangean relaxation $AJ(\alpha)$ of the following form:

$$\min \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j>i}}^{n} \sum_{k=1}^{n} \sum_{\substack{l=1 \\ l\neq k}}^{n} (c_{ijkl} - \alpha_{ikjl}) y_{ikjl}$$
$$- \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j<i}}^{n} \sum_{k=1}^{n} \sum_{\substack{l=1 \\ l\neq k}}^{n} (c_{ijkl} - \alpha_{jlik}) y_{ikjl} + \sum_{i=1}^{n} \sum_{k=1}^{n} a_{ik} b_{ik} x_{ik}$$

s.t.

$$\sum_{j=1}^{n} y_{ijkl} = x_{ik}, \quad 1 \leqslant i,k,l \leqslant n,$$

$$\sum_{l=1}^{n} y_{ijkl} = x_{ik}, \quad 1 \leqslant i,j,k \leqslant n,$$

$$0 \leqslant y_{ijkl} \leqslant 1, \quad 1 \leqslant i,j,k,l \leqslant n,$$

$$x_{ik} \in \mathbf{X}_n.$$

Let $\theta(\alpha)$ denote the optimal value of $AJ(\alpha)$. Then $\max_\alpha \theta(\alpha)$ equals the optimal value of the continuous relaxation of the Adams–Johnson linearization, see (30). Adams and Johnson show that for each fixed set of the multipliers α the problem $AJ(\alpha)$ can be solved efficiently by solving n^2+1 LAPs. Moreover they develop an iterative dual ascent procedure to solve approximately the above maximization problem which leads to the Adams–Johnson bound (AJB). This bound generalizes and unifies all previously mentioned bounds like GLB, the reduction bounds as well as the bound of Assad and Xu which can be obtained for special settings of the Lagrangean multipliers α_{ijkl}. It does not comprise, however, the bounds of Carraresi and Malucelli and the Hahn–Grant bound (HGB). Karisch et al. [92] showed recently that both, AJB and HGB can be obtained from the dual of the continuous relaxation of the MILP formulation (30) proposed by Adams and Johnson. They propose an iterative algorithm to solve this dual approximately and show that AJB, HGB, and all other Gilmore–Lawler-like bounds including the Carraresi–Malucelli bound can be obtained by applying this algorithm with specific settings for the control parameters. The same authors identify a setting for the parameters which seems to provide a bounding algorithm with a better time/quality trade-off than all the other mentioned bounding procedures.

6.3.2. Eigenvalue bounds

The trace formulation of a Koopmans–Beckmann problem can be used to derive a new class of bounds, the so-called *eigenvalue bounds* which were introduced by Finke et al. [68]. When implemented carefully, these techniques produce bounds of good quality in comparison to Gilmore–Lawler-like bounds. The eigenvalue bounds are, however, expensive in terms of computation time and deteriorate quickly when lower levels of a branch and bound tree are searched (see [50]).

Let us start from a Koopmans–Beckmann QAP with symmetric matrices A and B. In this case all eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of matrix A and $\mu_1, \mu_2, \ldots, \mu_n$ of matrix B are real. We collect these eigenvalues in vectors λ and μ, respectively. The matrices A and B possess diagonalizations of the form $A = P\Lambda P^{\mathrm{T}}$ and $B = QMQ^{\mathrm{T}}$ with orthogonal matrices P and Q and diagonal matrices $\Lambda = \mathrm{diag}(\lambda)$, $M = \mathrm{diag}(\mu)$. Since the columns $p_1, p_2, \ldots, p_n$ of P and $q_1, q_2, \ldots, q_n$ of Q form orthonormal bases, matrix S defined by $s_{ij} := \langle p_i, q_j \rangle^2$ is a doubly stochastic matrix which, due to Birkhoff's Theorem, can be written as convex combination of permutation matrices:

$$S = \sum_{\varphi \in \mathscr{S}_n} \alpha_\varphi X_\varphi.$$

It is easy to see that

$$tr\, AB = \lambda^{\mathrm{T}} S \mu = \sum_{\varphi \in \mathscr{S}_n} \alpha_\varphi \langle \lambda, X_\varphi \mu \rangle$$

which implies

$$\langle \lambda, \mu \rangle^- \leqslant tr\, AB \leqslant \langle \lambda, \mu \rangle^+.$$

Since XBX^{T} has the same eigenvalues as B, we get

Theorem 6.3 (Eigenvalue bounds for the QAP).

$$\langle \lambda, \mu \rangle^- \leqslant tr\, AXBX^{\mathrm{T}} \leqslant \langle \lambda, \mu \rangle^+.$$

By applying reduction techniques to the quadratic part of the objective function significant improvements can be achieved. Hadley et al. [81] consider also the case of non-symmetric QAPs and develop for them eigenvalue bounds by means of Hermitian matrices.

A more general approach to eigenvalue based lower bounding techniques was employed by Hadley et al. [79]. Consider the following sets of $n \times n$ matrices, where I is the $n \times n$ identity matrix and $\mathbf{1} := (1, \ldots, 1)^{\mathrm{T}}$ is the n-dimensional vector of all ones:

$$\begin{aligned} &\mathcal{O}_n := \{X : X^{\mathrm{T}}X = I\} && \text{set of } \textit{orthogonal } n \times n \textit{ matrices},\\ &\mathcal{E}_n := \{X : X\mathbf{1} = X^{\mathrm{T}}\mathbf{1} = \mathbf{1}\}, && \text{set of } n \times n \textit{ matrices with row}\\ &&& \textit{and column sums equal to one},\\ &\mathcal{N}_n := \{X : X \geqslant 0\}, && \text{set of } \textit{non-negative } n \times n \textit{ matrices}. \end{aligned} \tag{41}$$

It is a well known result that $\mathbf{X}_n = \mathcal{O}_n \cap \mathcal{E}_n \cap \mathcal{N}_n$. The above characterization of $\mathbf{X}_n$ implies that we get a relaxation of the QAP, if we delete one or two of the matrix sets $\mathcal{O}_n, \mathcal{E}_n$ and $\mathcal{N}_n$ in the intersection $\mathbf{X}_n = \mathcal{O}_n \cap \mathcal{E}_n \cap \mathcal{N}_n$. With respect to Theorem 6.3, Rendl and Wolkowicz [138] show that

$$\min_{X \in \mathcal{O}_n} tr(AXBX^{\mathrm{T}}) = \langle \lambda, \mu \rangle^-,$$

$$\max_{X \in \mathcal{O}_n} tr(AXBX^{\mathrm{T}}) = \langle \lambda, \mu \rangle^+.$$

In other words, the lower bound on the quadratic part of the QAP as obtained by the eigenvalue bound is derived by relaxing the feasible set to the set of orthogonal matrices.

A tighter relaxation was proposed in [80], where the set of permutation matrices was relaxed to $\mathcal{O}_n \cap \mathcal{E}_n$. The authors incorporate $\mathcal{E}_n$ in the objective function by exploiting the fact that for any $X \in \mathbf{X}_n$ the vector of ones is both a left and right eigenvector with eigenvalue 1.

The above considerations lead directly to bounds based on *semidefinite relaxations*. Let $\mathcal{Z}_n$ be the set of $n \times n$ 0–1 matrices. Then

$$\mathbf{X}_n = \mathcal{Z}_n \cap \mathcal{E}_n = \mathcal{Z}_n \cap \mathcal{O}_n.$$

Thus we can write a QAP in trace form as

$$\begin{aligned} &\min\ tr(AXBX^{\mathrm{T}} - CX^{\mathrm{T}})\\ &\text{s.t.}\\ &\quad XX^{\mathrm{T}} = X^{\mathrm{T}}X = I,\\ &\quad X\mathbf{1} = X^{\mathrm{T}}\mathbf{1} = \mathbf{1},\\ &\quad x_{ij}^2 - x_{ij} = 0. \end{aligned}$$

From this formulation semidefinite programming relaxations of the QAP can be obtained, see [154]. These relaxations are solved by interior point methods or cutting

plane methods. The quality of the bounds obtained in this way is competitive with the best existing lower bounds for the QAP. However, due to prohibitively high computation time requirements, the use of such approaches as basic bounding procedures within branch and bound algorithms is up to now not feasible.

6.4. Solution methods for QAPs

Since QAPs are notoriously hard to solve to optimality, there is a special need for good heuristics. It was a great surprise that even straightforward implementations of metaheuristics like *simulated annealing* (*SA*) or *tabu search* (*TS*) perform very well. An explanation for this behaviour will be given in Section 7.3.

The first simulated annealing algorithm for QAPs was published in Burkard and Rendl [40], soon after the first author heard a lecture of Černý [48] in Prague on the possibility to apply this thermodynamically motivated simulation procedure to the travelling salesman problem. (Černý recognized this possibility independently about at the same time as Kirkpatrick et al. [98].) Improved simulated annealing (SA) algorithms for the QAP have been proposed by several other authors, e.g. by Wilhelm and Ward [153] and Connolly [51].

Tabu search methods for the QAP have been proposed among others by Skorin-Kapov [143] and Taillard [146]. Taillard uses a so-called *robust tabu search* where the size of the tabu list is randomly chosen between a maximum and a minimum value. Battiti and Tecchiolli [14] developed the *reactive tabu search* which involves a mechanism for adopting the size of the tabu list. Reactive tabu search aims at improving the robustness of the algorithm. The algorithm notices when a cycle occurs, i.e., when a certain solution is revisited, and increases the tabu list size according to the length of the detected cycle. Computational results show that generally the reactive tabu search outperforms other tabu search algorithms for the QAP. More recently, parallel implementations of tabu search have been proposed, see e.g. [49]. Tabu search algorithms allow a natural parallel implementation by dividing the burden of the search in the neighborhood among several processors.

Among *genetic algorithms* for the QAP the approach due to Ahuja et al. [4] seems to outperform the others. It is a hybrid algorithm which combines features from greedy algorithms with ideas from genetic algorithms.

There is a number of codes available for solving QAPs. The reader is referred to QAPLIB, a library on quadratic assignment problems which is maintained by Burkard et al. [37]. QAPLIB contains programs, test instances with best known results and references with respect to QAPs and can be found at http://www.opt.math.tu-graz.ac.at/~karisch/qaplib. In particular FORTRAN codes for the GLB (up to the size $n=256$) and a branch and bound algorithm due to Burkard and Derigs [31] can be downloaded from the QAPLIB web page. Further one can find there the source file of a FORTRAN implementation of the simulated annealing algorithm of Burkard and Rendl [40]. Recently, Espersen, Karisch, Çela, and Clausen developed `QAPpack` which is a JAVA package containing a branch and bound algorithm to solve the QAP. `QAPpack` contains several different bounds and can be found at http://www.imm.dtu.dk/~te/QAPpack.

The source file of a C++ implementation of the simulated annealing algorithm of Connolly [51], due to Taillard, can be downloaded from Taillard's web page at . Also the source file of a PASCAL implementation of Taillard's robust tabu search algorithm can be found there.

Finally, the source file of a FORTRAN implementation of Li and Pardalos' generator for QAP instances with known optimal solution [107] can be obtained by sending an email to `coap@math.ufl.edu` with subject line `send 92006`.

6.5. Polynomially solvable special cases

Since QAPs are $\mathcal{NP}$-hard, the question arises, in which cases they can be solved in polynomial time. This means for Koopmans–Beckmann problems that the matrices A and B must have a special structure which enables a polynomial-time algorithm.

In contrast to the traveling salesman problem it turns out that the QAP with both coefficient matrices being Monge or Anti-Monge is $\mathcal{NP}$-hard, whereas the complexity of a QAP with one coefficient matrix being Monge and the other one being Anti-Monge is still open, see [26,47]. The case where A is a Monge matrix, B is a chess-board matrix and the size n of the problem is even, $n = 2m$, is solved by the permutation

$$\varphi(k) := \begin{cases} i, & \text{if} \quad k = 2i - 1,\ 1 \leqslant i \leqslant m, \\ m + i, & \text{if} \quad k = 2i,\ 1 \leqslant i \leqslant m. \end{cases}$$

The computational complexity in the case where n is odd, is open. Here a matrix $B = (b_{ij})$ is called a *chess-board matrix*, if its entries are given by $b_{ij} = (-1)^{i+j}$. A few other versions of the QAP involving Monge and Anti-Monge matrices with additional structural properties can be solved by dynamic programming.

Other special cases of the QAP involve matrices with a specific diagonal structure e.g. *circulant* and *Toeplitz matrices*. An $n \times n$ matrix $A = (a_{ij})$ is called a *Toeplitz matrix* if there exist numbers $c_{-n+1}, \ldots, c_{-1}, c_0, c_1, \ldots, c_{n-1}$ such that $a_{ij} = c_{j-i}$, for all i, j. A matrix A is called a *circulant matrix* if it is a Toeplitz matrix and the generating numbers c_i fulfill the conditions $c_i = c_{i-n}$, for $0 \leqslant i \leqslant n-1$. In other words, a Toeplitz matrix has constant entries along lines parallel to the diagonal, whereas a circulant is given by its first row and the entries of the ith row resembles the first row shifted circularly by $i - 1$ places to the right.

QAPs with one Anti-Monge (Monge) matrix and one Toeplitz (circulant) matrix remain $\mathcal{NP}$-hard unless additional conditions are imposed on the coefficient matrices. A well studied problem is the so called Anti-Monge–Toeplitz QAP where the rows and columns of the Anti-Monge matrix are non-decreasing, see [29]. It has been shown that this problem is $\mathcal{NP}$-hard and contains as a special case the so called *turbine runner problem* introduced by Mosewich [114] and formulated as a QAP by Laporte and Mercure [102]. In the turbine runner problem we are given n blades to be welded in regular spacing around the cylinder of the turbine. Due to inaccuracies in the manufacturing process the masses m_i of the blades differ slightly and consequently the gravity center of the system does not lie on the rotation axis of the cylinder, leading to instabilities. In an effort to make the system as stable as possible, it is desirable to

locate the blades so as to minimize the distance between the center of gravity and the rotation axis. Mathematically, this problem can be formulated as QAP of the form

$$\min_{\varphi} \sum_{i=1}^{n} \sum_{j=1}^{n} \cos\left(\frac{2(i-j)\pi}{n}\right) m_{\varphi(i)} m_{\varphi(j)}.$$

Note that matrix A with $a_{ij} = \cos(2(i-j)\pi/n)$ is a periodic Toeplitz matrix, whereas matrix $B=(b_{kl})$ with $b_{kl} := m_k m_l$ is an Anti-Monge matrix, if the masses m_k are sorted decreasingly. It turns out that the *maximization* version of this problem is polynomially solvable, whereas the minimization of the objective function (6.5) is $\mathcal{NP}$-hard.

Further polynomially solvable special cases of the Anti-Monge–Toeplitz QAP arise if additional constraints e.g. *benevolence* or *k-benevolence* are imposed on the Toeplitz matrix. These conditions are expressed in terms of properties of the generating function of these matrices, see [29].

All polynomially solvable QAPs described above, where A is an Anti-Monge (Monge) matrix and B is a Toeplitz (circulant) matrix are *constant permutation QAPs.* This means that an optimal solution can be specified explicitly. The technique used to prove this fact and to identify the optimal permutation is called *reduction to extremal rays*. This technique exploits two facts: first, the involved matrix classes form cones, and secondly, the objective function of the QAP is linear with respect to each of the coefficient matrices. These two facts allow us to restrict the investigations to instances of the QAP with 0–1 coefficient matrices which are extremal rays of the above mentioned cones.

The identification of polynomially solvable special cases of the QAP which are not constant permutation QAPs and can be solved algorithmically remains a challenging open question.

A subclass of Monge matrices are the so-called *Kalmanson matrices.* A matrix $A=(a_{ij})$ is a *Kalmanson matrix*, if it is symmetric and its entries satisfy the following inequalities for all indices $i,j,k,l,\ i<j<k<l$:

$$a_{ij} + a_{kl} \leqslant a_{ik} + a_{jl}, \quad a_{il} + a_{jk} \leqslant a_{ik} + a_{jl}.$$

(For more information on Monge, Anti-Monge and Kalmanson matrices, and their properties the reader is referred to the survey article of Burkard et al. [38].) The QAP(A,B) with a Kalmanson matrix A and a Toeplitz matrix B has been investigated by Deĭneko and Woeginger [55]. The computational complexity of this problem is an open question, but analogously as in the case of the Anti-Monge–Toeplitz QAP, polynomially solvable versions of the problem are obtained by imposing additional constraints to the Toeplitz matrix.

7. Asymptotic results for assignment problems

Assignment problems show an interesting behaviour when their size tends to infinity. Whereas for linear assignment problems the gap between best and worst solution tends to infinity as the problem size increases, the best and the worst solution of quadratic

assignment problems tend almost surely to the same value, when the size of the problems increases. Let us describe first what is known about the asymptotic behaviour of linear sum assignment problems. Throughout this section we assume that the cost coefficients of the problems are independent random variables with a common prespecified distribution.

7.1. Asymptotic results for linear sum assignment problems

Closely related with the question about the expected behaviour of the optimal value of a linear sum assignment problem is the question whether a random bipartite graph admits a perfect matching or not. The existence of a perfect matching in a bipartite graph G is intuitively connected with the number of edges of G and the fact that G does not contain isolated vertices. Walkup [152] considers the class $\mathcal{G}(n,d)$ of *directed* bipartite graphs $G=(V,W;E)$ with $|V|=|W|=n$, where each vertex has out-degree d. Let $P(n,d)$ be the probability that a graph chosen randomly from $\mathcal{G}(n,d)$ contains a perfect matching. Walkup shows that $P(n,1)$ tends to 0, but for all $d \geqslant 2$, $P(n,d)$ tends to 1 as n approaches infinity. (Notice that due to the Marriage Theorem 2.1 the existence of a perfect matching in an undirected bipartite graph, regular of degree d, is trivial.) Using the above result Walkup [151] shows in a following paper that 3 is an upper bound on the expected optimal value of the LSAP in the case that the cost coefficients c_{ij} are independent random variables uniformly distributed on $[0,1]$.

Four years later Karp [95] improved the upper bound on the optimum objective function value of an LSAP to 2. Both, Walkup's and Karp's proofs are non-constructive and cannot be exploited in heuristics for producing assignments with expected optimal value within the given bounds.

Independent and uniformly distributed cost elements c_{ij} on $[0,1]$ lead immediately to independent and uniformly distributed cost elements $\bar{c}_{ij} := 1 - c_{ij}$ on $[0,1]$. Therefore we can derive from the following equality

$$\max_{\varphi} \sum_{i=1}^{n} \bar{c}_{i\varphi(i)} = n - \min_{\varphi} \sum_{i=1}^{n} c_{i\varphi(i)}$$

that the maximum objective function value of a linear assignment problem tends to infinity as the problem size increases. Thus the gap between minimum and maximum objective function values of a LSAP becomes arbitrarily large, when the problem size increases.

Lower bounds for the expected optimal value of the LSAP with independent and uniformly distributed costs c_{ij} on $[0,1]$ are given by Lazarus [104]. The author exploits weak duality and evaluates the expected value of the dual objective function $\sum_i u_i + \sum_j v_j$ achieved after row and column reductions, see the algorithm in Section 4.1. By computations involving first order statistics it can be shown that the expected value of $\sum_i u_i + \sum_j v_j$—which is a lower bound for the expected optimal value of the LSAP—is of order $1 + 1/e + \log n/n$. This yields a bound of 1.368. Moreover, Lazarus evaluates the maximum number of 0-entries in the cost matrix

after row and column reductions. It turns out that the probability of finding an optimal assignment only after row and column reductions tends to 0 as n tends to infinity.

The lower bound on the expected optimal value of an LSAP was improved by Olin [120] to 1.51, which is currently the best value known. Olin considers first the solution of the dual obtained by row and column reductions and improves this solution of the dual by adding the second smallest element of each row to all elements in that row. Then the largest among the terms added to the elements of a column is subtracted from this column. This transformation leads to a new dual feasible solution which yields an expected value of the objective function equal to 1.47. By applying an analogous transformation starting from the columns of the reduced cost matrix, the bound is increased to 1.51. Mézard and Parisi [113] conjectured that the expected optimal value of an LSAP is $\pi^2/6 = 1.645$, if the cost coefficients are independent and uniformly distributed random variables in $[0,1]$. Indeed, Donath [61] observed in his computational experiments a value close to 1.6. For further asymptotic results concerning different distribution functions of the cost coefficients c_{ij} see the recent handbook article of Burkard and Çela [24].

7.2. Asymptotic results for linear bottleneck assignment problems

Pferschy [125] investigates the asymptotic behavior of linear bottleneck assignment problems. He shows that the expected value of an optimal solution of LBAP tends towards the lower end of the range of cost coefficients for any bounded distribution function when the size n of the problem increases. In particular he shows:

Theorem 7.1. *If* $\sup\{x|F(x) < 1\} < \infty$, *then the optimal solution* Z_n *of a random LBAP with cost coefficients distributed according to distribution function* F *satisfies*

$$\lim_{n\to\infty} E[Z_n] = \inf\{x|F(x) > 0\}.$$

In the case of uniformly distributed cost coefficients in $[0,1]$ Pferschy derives the following lower and upper bounds for $E(Z_n)$.

Theorem 7.2. *Let* $B(x,y)$ *be the Beta function. Then we get for* $n > 78$:

$$E[Z_n] < 1 - \left[\frac{2}{n(n+2)}\right]^{2/n} \frac{n}{n+2} + \frac{123}{610n}$$

and

$$E[Z_n] \geqslant 1 - nB\left(n, 1+\frac{1}{n}\right) = \frac{\ln n + 0.5749}{n} + \mathrm{O}\left(\frac{\ln^2 n}{n^2}\right).$$

7.3. Asymptotic results for quadratic sum assignment problems

In contrast to linear assignment problems it can be shown that under mild probabilistic assumptions the ratio between "best" and "worst" values of the objective function of a QAP approaches 1 as the size of the problem tends to infinity. This is a very strange and interesting asymptotic behaviour which was at first proved for QAPs by Burkard and Fincke [33] in 1983. Later, in 1985, a whole class of combinatorial optimization problems showing this behaviour was found, see Burkard and Fincke [34]. As a consequence of this behaviour we can expect that *every heuristic finds an almost optimal solution when applied to QAP instances which are large enough.* On the other hand this behaviour shows that the landscape of objective function values for QAPs is very flat. This implies that it will be difficult for branch and bound procedures to detect the true optimal solution. Indeed, Dyer et al. [62] showed the following result for QAPs whose coefficient matrices have independently distributed random entries with a common distribution.

Theorem 7.3 (Dyer et al. [62]). *Consider any branch and bound algorithm for solving a QAP with randomly generated coefficients which have finite expected values, variances and third moments. Assume that the branch and bound algorithm assigns one index in each step and employs a Gilmore–Lawler bound. Then the number of branched nodes explored is at least* $n^{(1-o(1))n/4}$ *with a probability tending to* 1 *as the size* n *of the QAP tends to infinity.*

Burkard and Fincke [33] investigate the relative difference between the worst and the best value of the objective function for Koopmans–Beckmann QAPs. First the Euclidean case is considered, where A is the distance matrix of independently and uniformly distributed points in the unit square. Then they consider the general case where the entries of matrices A and B are independent random variables taken from a uniform distribution on $[0,1]$. In both cases it is shown that the relative difference between the best and worst solution values approaches 1 with a probability tending to 1 as the size of the problem tends to infinity. These results were strengthened by Frenk et al. [70] as well as by Rhee [139]. Their results can be summarized in the following theorem:

Theorem 7.4 (Frenk et al. [70] and Rhee [139]). *Consider a sequence of QAPs* $(A^{(n)}, B^{(n)})$ *whose coefficients are independently distributed random variables in* $[0,M]$ *with expected values* $E(A)$ *and* $E(B)$, *resp., all entries of* A (*and* B, *resp.*) *having the same distribution. Denote by* $Z(A^{(n)}, B^{(n)}, \varphi)$ *the value of the QAP with respect to permutation* φ. *Then there exists a constant* K_1 (*which does not depend on* n), *such that the following inequality holds almost surely*:

$$\limsup_{n\to\infty} \frac{\sqrt{n}}{\sqrt{\log n}} \left| \frac{Z(A^{(n)}, B^{(n)}, \varphi)}{n^2 E(A)E(B)} - 1 \right| \leqslant K_1.$$

Moreover, let Y *be a random variable defined by*

$$Y = Z(A^{(n)}, B^{(n)}, \varphi^{(n)}_{\mathrm{opt}}) - n^2 E(A)E(B),$$

where $\varphi_{\mathrm{opt}}^{(n)}$ *is an optimal solution of* $QAP(A^{(n)}, B^{(n)})$. *Then there exists another constant* K_2, *also independent of the size of the problem, such that*

$$\frac{1}{K_2} n^{3/2} (\log n)^{1/2} \leqslant E(Y) \leqslant K_2 n^{3/2} (\log n)^{1/2},$$

$$P\{|Y - E(Y)| \geqslant t\} \leqslant 2 \exp\left(\frac{-t^2}{4n^2 ||A||_\infty^2 ||B||_\infty^2} \right)$$

for each $t \geqslant 0$, *where* $E(Y)$ *denotes the expected value of variable* Y *and* $||A||_\infty$ ($||B||_\infty$) *is the so-called* row sum norm of *matrix* A (B) *defined by* $||A||_\infty = \max_{1 \leqslant i \leqslant n} \sum_{j=1}^{n} |a_{ij}|$.

Later it was shown by Burkard and Fincke [34] that this strange asymptotic behaviour is related to the ratio between the number of (non-zero) cost coefficients and the logarithm of the number of feasible solutions of a combinatorial optimization problem. Consider a sequence P_n of combinatorial optimization (minimization) problems defined on finite ground sets E_n. A feasible solution of the problem can be represented by a subset F of E_n. Let $\mathscr{F}_n$ denote the class of all feasible solutions of problem P_n. Further let $c_n : E_n \to \mathbb{R}^+$ be the cost coefficients and $z_n(F) := \sum_{e \in F} c_n(e)$ be the objective function value of a feasible solution F. Then the following theorem can be shown:

Theorem 7.5 (Burkard and Fincke [34], Szpankowski [144] and Burkard and Çela [23]). *Consider a sequence of combinatorial optimization problems* P_n *with the following properties*:

- *All feasible solutions of problem* P_n *have the same cardinality* $|F_n|$.
- *Every element* $e \in E_n$ *occurs in the same number of feasible solutions* F_n.
- *The cost elements* $c_n(e)$, $e \in E_n$ *are i.i.d. random variables in* $[0, M]$.
- $\lim_{n \to \infty} \frac{\log |\mathscr{F}_n|}{|F_n|} = 0$.

Then all *solution values converge almost surely to n times the expected value of the cost coefficients* $c_n(e)$, *i.e., all solutions have asymptotically almost surely the same value.*

This theorem was first proven in 1985 by Burkard and Fincke [34] in a weaker form, namely assuring convergence in probability. The authors gave, however, explicit bounds for the asymptotic behaviour. Szpankowski [144] showed 1995 the convergence almost surely. Recently, Burkard and Çela [23] derived this theorem from thermodynamical considerations using the Boltzmann distribution. This is of particular interest, since simulated annealing is a simulation tool, which yields excellent results for these problems and stems from the same background. Note that the condition

$$\lim_{n \to \infty} \frac{\log |\mathscr{F}_n|}{|F_n|} = 0$$

is fulfilled for QAPs, but not for LAPs. Both problems have $|\mathscr{F}_n| = n!$ feasible solutions, but a feasible solution of a LAP has n coefficients, whereas a feasible solution of a QAP has n^2 coefficients in the objective function.

7.4. Asymptotic results for quadratic bottleneck assignment problems

Similar results as in the sum case hold also for quadratic bottleneck assignment problems. Let $z_{\mathrm{opt}}(n)$ be the minimum objective function value of a BQAP of size n and let z_{wor} be the maximum objective function value. In Burkard and Fincke [32] the following theorem has been proven:

Theorem 7.6. *Let a_{ij} and b_{kl} be independent and uniform* $[0,1]$ *random variables for* $1 \leqslant i,j,k,l \leqslant n$. *Then*

$$\lim_{n\to\infty} \mathbf{P}\left\{ \frac{z_{\mathrm{wor}} - z_{\mathrm{opt}}}{z_{\mathrm{opt}}} \leqslant \left(\left(\frac{n}{2\log n} \right)^{1/2} - 1 \right)^{-1} \right\} = 1.$$

In the case of general combinatorial optimization problems it has been shown by Burkard and Fincke [34] that this asymptotic behaviour relies again on the ratio of the logarithm of the number of feasible solutions to the number of non-zero cost coefficients in a feasible solution. If this ratio tends to 0 for increasing problem sizes then all solutions have almost the same value in probability.

8. Further assignment models

8.1. Biquadratic assignment problems

In connection with VLSI synthesis programmable logic arrays (PLA) have to be implemented. We want to find an encoding of states such that the actual implementation by flip-flops is of minimum size. In the case of data flip-flops this leads to a quadratic assignment problem. If, however, toggle flip-flops are used, a new problem arises which is called *biquadratic assignment problem (BiQAP)*. Mathematically we can formulate a BiQAP as follows. Let two arrays $A = (a_{ijkl})$ and $B = (b_{mpst})$ with $1 \leqslant i,j,k,l,m,p,s,t \leqslant n$ be given. The BiQAP can then be stated as:

$$\min_{\varphi\in\mathscr{S}_n} \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} a_{ijkl} b_{\varphi(i)\varphi(j)\varphi(k)\varphi(l)}.$$

Burkard et al. [27] compute lower bounds for the BiQAP derived from lower bounds of the QAP. The computational results show that these bounds are weak and deteriorate as the size of the problem increases. This observation suggests that branch and bound methods will only be effective on very small instances. For larger instances efficient heuristics that find good-quality approximate solutions are needed. Several heuristics for the BiQAP have been developed by Burkard and Çela [21], in particular deterministic improvement methods and variants of simulated annealing and tabu search algorithms. Computational experiments on test problems of size up to $n = 32$, for which the optimum objective function value is known beforehand, suggest that a specific simulated annealing algorithm is best suited for this kind of problems.

BiQAPs show the same asymptotic behaviour as QAPs as has been shown and analyzed in detail in Burkard et al. [27].

8.2. Communication assignment problems

Let us consider n communication centers $C_1, C_2, \ldots, C_n$ which have to be assigned to the vertices of an undirected, connected graph. Each communication center C_i transmits messages to every other center C_j at a rate of t_{ij} messages per time unit. In case that there is no direct connection between the centers C_i and C_j the messages are routed via intermediate centers. Once the communication centers are assigned to the vertices of the graph, the messages have to be routed. We distinguish between two different routing patterns: In the *single path model* for every pair of communication centers a single path in the graph is selected and the whole traffic is routed along this path. In the *fractional model* the traffic is split into several parts which are routed along different paths from the origin to the destination. For any fixed assignment φ and any fixed routing pattern ρ we denote the overall amount of traffic passing through center C_i as $N(C_i; \varphi, \rho)$ and call this the *noise* at center C_i.

The communication assignment problem asks for an assignment of the communication centers to the vertices of the underlying graph and for a routing pattern such that the maximum noise becomes as small as possible:

$$\min_{\varphi,\rho} \max_{1 \leqslant i \leqslant n} N(C_i; \varphi, \rho).$$

In Burkard et al. [30] it has been shown that the communication assignment problem is $\mathcal{NP}$-hard even for such simple graphs as paths, cycles and star graphs with branch length 3. By using a perfect matching algorithm and binary search the communication assignment problem can be solved in polynomial time in star graphs of branch length $\leqslant 2$. Further it has been shown that in so-called double star graphs the single path version is $\mathcal{NP}$-hard, whereas the fractional model can be solved in polynomial time by a sequence of $O(n^2)$ linear programs with $n-2$ variables and n^2+n constraints each. Moreover, the authors develop a branch and bound algorithm for solving this problem in trees. Later, Burkard et al. [25] designed various heuristics like simulated annealing and tabu search for this problem. The good performance of these heuristics seems again be related to the asymptotic behaviour of communication assignment problems. These problems show again a similar asymptotic behaviour as QAPs as can be seen in the following theorem:

Theorem 8.1. *Consider a sequence of communication assignment problem instances whose entries t_{ij} are i.i.d. random variables on a compact interval $[0, M]$. Then the ratio between maximum noise and minimum noise approaches 1 with a probability tending to 1 as the size of the problems tends to infinity.*

References

[1] W.P. Adams, T.A. Johnson, Improved linear programming-based lower bounds for the quadratic assignment problem, in: P.M. Pardalos, H. Wolkowicz (Eds.), Quadratic Assignment and Related

Problems, DIMACS Series on Discrete Mathematics and Theoretical Computer Science, Vol. 16, AMS, Providence, RI, 1994, pp. 43–75.
[2] W.P. Adams, H.D. Sherali, A tight linearization and an algorithm for zero-one quadratic programming problems, Management Sci. 32 (1986) 1274–1290.
[3] W.P. Adams, H.D. Sherali, Linearization strategies for a class of zero-one mixed integer programming problems, Oper. Res. 38 (1990) 217–226.
[4] R.K. Ahuja, J.B. Orlin, A. Tivari, A descent genetic algorithm for the quadratic assignment problem, Computers and Operations Research 27 (2000) 917–934.
[5] H. Alt, N. Blum, K. Mehlhorn, M. Paul, Computing maximum cardinality matching in time $O(n^{1.5}\sqrt{m/\log n})$, Inform. Process. Lett. 37 (1991) 237–240.
[6] S. Arora, A. Frieze, H. Kaplan, A new rounding procedure for the assignment problem with applications to dense graph arrangement problems, Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 1996, pp. 21–30.
[7] A.A. Assad, W. Xu, On lower bounds for a class of quadratic 0–1 programs, Oper. Res. Lett. 4 (1985) 175–180.
[8] E. Balas, P.R. Landweer, Traffic assignment in communications satellites, Oper. Res. Lett. 2 (1983) 141–147.
[9] E. Balas, L. Qi, Linear-time separation algorithms for the three-index assignment polytope, Discrete Appl. Math. 43 (1993) 1–12.
[10] E. Balas, M.J. Saltzman, Facets of the three-index assignment polytope, Discrete Appl. Math. 23 (1989) 201–229.
[11] E. Balas, M.J. Saltzman, An algorithm for the three-index assignment problem, Oper. Res. 39 (1991) 150–161.
[12] S.E. Bammel, J. Rothstein, The number of 9×9 Latin squares, Discrete Math. 11 (1975) 93–95.
[13] A.I. Barvinok, Computational complexity of orbits in representations of symmetric groups, Adv. Soviet Math. 9 (1992) 161–182.
[14] R. Battiti, G. Tecchiolli, The reactive tabu search, ORSA J. Comput. 6 (1994) 126–140.
[15] G. Birkhoff, Tres observaciones sobre el algebra lineal, Rev. univ. nac. Tucumán (A) 5 (1946) 147–151.
[16] W.L. Brogan, Algorithm for ranked assignments with applications to multiobject tracking, J. Guidance 12 (1989) 357–364.
[17] R.E. Burkard, Die Störungsmethode zur Lösung quadratischer Zuordnungsprobleme, Oper. Res. Verfahren 16 (1973) 84–108.
[18] R.E. Burkard, Quadratische Bottleneckprobleme, Oper. Res. Verfahren 18 (1974) 26–41.
[19] R.E. Burkard, Time-slot assignment for TDMA-systems, Computing 35 (1985) 99–112.
[20] R.E. Burkard, Locations with spatial interactions: the quadratic assignment problem, in: P.B. Mirchandani, R.L. Francis (Eds.), Discrete Location Theory, Wiley, New York, 1991, pp. 387–437.
[21] R.E. Burkard, E. Çela, Heuristics for biquadratic assignment problems and their computational comparison, European J. Oper. Res. 83 (1995) 283–300.
[22] R.E. Burkard, E. Çela, Quadratic and three-dimensional assignments, in: M. Dell'Amico, F. Maffioli, S. Martello (Eds.), Annotated Bibliographies in Combinatorial Optimization, Wiley, Chichester, 1997, pp. 373–391.
[23] R.E. Burkard, E. Çela, An asymptotical study of combinatorial optimization problems by means of statistical mechanics. SFB Report 133, Institute of Mathematics, Technical University Graz, Austria, 1998.
[24] R.E. Burkard, E. Çela, Linear assignment problems and extensions, in: D.-Z. Du, P.M. Pardalos (Eds.), Handbook of Combinatorial Optimization, Vol. 4, Kluwer Academic Publishers, Dordrecht, 1999.
[25] R.E. Burkard, E. Çela, T. Dudás, A communication assignment problem on trees: heuristics and asymptotic behaviour, in: P.M. Pardalos, D.W. Hearn, W.W. Hager (Eds.), Network Optimization, Lecture Notes in Economics and Mathematical Systems, Vol. 450, Springer, Berlin, 1997, pp. 127–155.
[26] R.E. Burkard, E. Çela, V.M. Demidenko, N.N. Metelski, G.J. Woeginger, Perspectives of easy and hard cases of the quadratic assignment problems, SFB Report 104, Institute of Mathematics, Technical University Graz, Austria, 1997.

[27] R.E. Burkard, E. Çela, B. Klinz, On the biquadratic assignment problem, in: P.M. Pardalos, H. Wolkowicz (Eds.), Quadratic Assignment and Related Problems, DIMACS Series on Discrete Mathematics and Theoretical Computer Science, Vol. 16, AMS, Providence, RI, 1994, pp. 117–146.
[28] R.E. Burkard, E. Çela, P. Pardalos, L.S. Pitsoulis, The quadratic assignment problem, in: D.-Z. Du, P.M. Pardalos (Eds.), Handbook of Combinatorial Optimization, Vol. 3, Kluwer Academic Publishers, Dordrecht, 1998, pp. 241–339.
[29] R.E. Burkard, E. Çela, G. Rote, G.J. Woeginger, The quadratic assignment problem with an Anti-Monge and a Toeplitz matrix: easy and hard cases, Math. Programming 82 (1998) 125–158.
[30] R.E. Burkard, E. Çela, G.J. Woeginger, A minimax assignment problem in treelike communication networks, European J. Oper. Res. 87 (1995) 670–684.
[31] R.E. Burkard, U. Derigs, Assignment and Matching Problems: Solution Methods with FORTRAN Programs, Springer, Berlin, 1980.
[32] R.E. Burkard, U. Fincke, On random quadratic bottleneck assignment problems, Math. Programming 23 (1982) 227–232.
[33] R.E. Burkard, U. Fincke, The asymptotic probabilistic behavior of the quadratic sum assignment problem, Z. Oper. Res. 27 (1983) 73–81.
[34] R.E. Burkard, U. Fincke, Probabilistic asymptotic properties of some combinatorial optimization problems, Discrete Appl. Math. 12 (1985) 21–29.
[35] R.E. Burkard, K. Fröhlich, Some remarks on 3-dimensional assignment problems, Methods of Oper. Res. 36 (1980) 31–36.
[36] R.E. Burkard, W. Hahn, U. Zimmermann, An algebraic approach to assignment problems, Math. Programming 12 (1977) 318–327.
[37] R.E. Burkard, S.E. Karisch, F. Rendl, QAPLIB—a quadratic assignment problem library, J. Global Optim. 10 (1997) 391–403. An on-line version is available via World Wide Web at the following URL: http://www.opt.math.tu-graz.ac.at/∼karisch/qaplib/
[38] R.E. Burkard, B. Klinz, R. Rudolf, Perspectives of Monge properties in optimization, Discrete Appl. Math. 70 (1996) 95–161.
[39] R.E. Burkard, J. Offermann, Entwurf von Schreibmaschinentastaturen mittels quadratischer Zuordnungsprobleme, Z. Oper. Res. 21 (1977) B121–B132 (in German).
[40] R.E. Burkard, F. Rendl, A thermodynamically motivated simulation procedure for combinatorial optimization problems, European J. Oper. Res. 17 (1984) 169–174.
[41] R.E. Burkard, R. Rudolf, Computational investigations on 3-dimensional axial assignment problems, Belgian J. Oper. Res. 32 (1993) 85–98.
[42] R.E. Burkard, R. Rudolf, G.J. Woeginger, Three dimensional axial assignment problems with decomposable cost coefficients, Discrete Appl. Math. 65 (1996) 123–169.
[43] R.E. Burkard, U. Zimmermann, Weakly admissible transformations for solving algebraic assignment and transportation problems, Math. Programming Study 12 (1980) 1–18.
[44] R.E. Burkard, U. Zimmermann, Combinatorial optimization in linearly ordered semimodules: a survey, in: B. Korte (Ed.), Modern Applied Mathematics, North-Holland, Amsterdam, 1982, pp. 392–436.
[45] G. Carpaneto, P. Toth, Algorithm for the solution of the bottleneck assignment problem, Computing 27 (1981) 179–187.
[46] P. Carraresi, F. Malucelli, A reformulation scheme and new lower bounds for the QAP, in: P. Pardalos, H. Wolkowicz (Eds.), Quadratic Assignment and Related Problems, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 16, AMS, Providence, RI, 1994, pp. 147–160.
[47] E. Çela, The Quadratic Assignment Problem: Theory and Algorithms, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.
[48] V. Černý , Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm, J. Optim. Theory Appl. 45 (1985) 41–51.
[49] J. Chakrapani, J. Skorin-Kapov, Massively parallel tabu search for the quadratic assignment problem, Ann. Oper. Res. 41 (1993) 327–342.
[50] J. Clausen, S.E. Karisch, M. Perregaard, F. Rendl, On the applicability of lower bounds for solving rectilinear quadratic assignment problems in parallel, Comput. Optim. Appl. 10 (1998) 127–147.
[51] D.T. Connolly, An improved annealing scheme for the QAP, European J. Oper. Res. 46 (1990) 93–100.

[52] K. Conrad, Das Quadratische Zuweisungsproblem und zwei seiner Spezialfälle, Mohr-Siebeck, Tübingen, 1971.
[53] D. Coppersmith, S. Vinograd, Matrix multiplication via arithmetic progressions, J. Symbolic Comput. 9 (1990) 251–280.
[54] V.G. Deĭneko, V.L. Filonenko, On the reconstruction of specially structured matrices, Aktualnyje Problemy EVM i Programmirovanije, Dnjepropetrovsk, DGU, 1979 (1990) 251–280 (in Russian).
[55] V.G. Deĭneko, G.J. Woeginger, A solvable case of the quadratic assignment problem, Oper. Res. Lett. 22 (1998) 13–17.
[56] M. Dell'Amico, S. Martello, Linear Assignment, in: M. Dell'Amico, F. Maffioli, S. Martello (Eds.), Annotated Bibliographies in Combinatorial Optimization, Wiley, Chichester, 1997, pp. 355–371.
[57] U. Derigs, Alternate strategies for solving bottleneck assignment problems—analysis and computational results, Computing 33 (1984) 95–106.
[58] U. Derigs, O. Goecke, R. Schrader, Monge sequences and a simple assignment algorithm, Discrete Appl. Math. 15 (1986) 241–248.
[59] U. Derigs, U. Zimmermann, An augmenting path method for solving linear bottleneck assignment problems, Computing 19 (1978) 285–295.
[60] J.W. Dickey, J.W. Hopkins, Campus building arrangement using TOPAZ, Transportation Res. 6 (1972) 59–68.
[61] W.E. Donath, Algorithms and average-value bounds for assignment problems, IBM J. Res. Develop. 13 (1969) 380–386.
[62] M.E. Dyer, A.M. Frieze, C.J.H. McDiarmid, On linear programs with random costs, Math. Programming 35 (1986) 3–16.
[63] J. Edmonds, D.R. Fulkerson, Bottleneck extrema, J. Combin. Theory 8 (1970) 299–306.
[64] C.S. Edwards, A branch and bound algorithm for the Koopmans–Beckmann quadratic assignment problem, Math. Programming Stud. 13 (1980) 35–52.
[65] R. Euler, Odd cycles and a class of facets of the axial 3-index assignment polytope, Applicationes Math. (Zastowania Matematyki) 19 (1987) 375–386.
[66] R. Euler, R.E. Burkard, R. Grommes, On Latin squares and the facial structure of related polytopes, Discrete Math. 62 (1986) 155–181.
[67] R. Euler, H. Le Verge, Time-tables, polyhedra and the greedy algorithm, Discrete Appl. Math. 65 (1996) 207–221.
[68] G. Finke, R.E. Burkard, F. Rendl, Quadratic assignment problems, Ann. Discrete Math. 31 (1987) 61–82.
[69] L.R. Ford, D.R. Fulkerson, Maximal flow through a network, Canad. J. Math. 8 (1956) 399–404.
[70] J.B.G. Frenk, M. van Houweninge, A.H.G. Rinnooy Kan, Asymptotic properties of the quadratic assignment problem, Math. Oper. Res. 10 (1985) 100–116.
[71] A.M. Frieze, Complexity of a 3-dimensional assignment problem, European J. Oper. Res. 13 (1983) 161–164.
[72] A.M. Frieze, L. Yadegar, An algorithm for solving 3-dimensional assignment problems with application to scheduling in a teaching practice, J. Oper. Res. Soc. 32 (1981) 989–995.
[73] A.M. Frieze, J. Yadegar, On the quadratic assignment problem, Discrete Appl. Math. 5 (1983) 89–98.
[74] R. Fulkerson, I. Glicksberg, O. Gross, A production line assignment problem, Technical Report RM-1102, The Rand Corporation, Sta. Monica, CA, 1953.
[75] H.N. Gabow, R.E. Tarjan, Algorithms for two bottleneck optimization problems, J. Algorithms 9 (1988) 411–417.
[76] R. Garfinkel, An improved algorithm for the bottleneck assignment problem, Oper. Res. 19 (1971) 1747–1751.
[77] A.M. Geoffrion, G.W. Graves, Scheduling parallel production lines with changeover costs: Practical applications of a quadratic assignment/LP approach, Oper. Res. 24 (1976) 595–610.
[78] O. Gross, The bottleneck assignment problem, Technical Report P-1630, The Rand Corporation, Sta. Monica, CA, 1959.
[79] S.W. Hadley, F. Rendl, H. Wolkowicz, Bounds for the quadratic assignment problem using continuous optimization techniques, Proceedings of the First Integer Programming and Combinatorial Optimization Conference (IPCO), University of Waterloo Press, 1990, pp. 237–248.

[80] S.W. Hadley, F. Rendl, H. Wolkowicz, A new lower bound via projection for the quadratic assignment problem, Math. Oper. Res. 17 (1992) 727–739.
[81] S.W. Hadley, F. Rendl, H. Wolkowicz, Nonsymmetric quadratic assignment problems and the Hoffman-Wielandt inequality, Linear Algebra Appl. 58 (1992) 109–124.
[82] P. Hahn, T. Grant, Lower bounds for the quadratic assignment problem based upon a dual formulation, Oper. Res. 46 (1998) 912–922.
[83] Ph. Hall, On representatives of subsets, J. London Math. Soc. 10 (1935) 26–30.
[84] P. Hansen, L. Kaufman, A primal-dual algorithm for the three-dimensional assignment problem, Cahiers du CERO 15 (1973) 327–336.
[85] G.H. Hardy, J.E. Littlewood, G. Pólya, Inequalities, Cambridge University Press, London, 1952.
[86] D.R. Heffley, Assigning runners to a relay team, in: S.P. Ladany, R.E. Machol (Eds.), Optimal Strategies in Sports, North-Holland, Amsterdam, 1977, pp. 169–171.
[87] A.J. Hoffman, On simple linear programming problems, in: V. Klee (Ed.), Convexity, Proceedings of Symposia in Pure Mathematics, Vol. 7, AMS, Providence, RI, 1963, pp. 317–327.
[88] J.E. Hopcroft, R.M. Karp, An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, SIAM J. Comput. 2 (1973) 225–231.
[89] M. Jünger, V. Kaibel, The QAP polytope, and the star transformation, Discr. Appl. Math. 111 (2001) 283–306.
[90] M. Jünger, V. Kaibel, On the SQAP polytope, SIAM J. Optim. 11 (2000) 444–463.
[91] V. Kaibel, Polyhedral Combinatorics of the Quadratic Assignment Problem, Ph.D. Thesis, Universität zu Köln, Germany, 1997.
[92] S.E. Karisch, E. Çela, J. Clausen, T. Espersen, A dual framework for lower bounds of the quadratic assignment problem based on linearization, Computing 63 (1999) 351–403.
[93] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher (Eds.), Complexity of Computer Computations, Plenum Press, New York, 1972, pp. 85–103.
[94] R.M. Karp, An algorithm to solve the $m \times n$ assignment problem in expected time $O(mn \log n)$, Networks 10 (1980) 143–152.
[95] R.M. Karp, An upper bound on the expected cost of an optimal assignment, in: D.S. Johnson, T. Nishizeki, A. Nozaki, H.S. Wilf (Eds.), Perspectives in Computing, Vol. 15, Academic Press, Boston, 1987, pp. 1–4.
[96] R.M. Karp, A.H.G. Rinnooy Kan, R.V. Vohra, Average case analysis of a heuristic for the assignment problem, Math. Oper. Res. 19 (1994) 513–522.
[97] L. Kaufman, F. Broeckx, An algorithm for the quadratic assignment problem using Benders' decomposition, European J. Oper. Res. 2 (1978) 204–211.
[98] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[99] D. König, Graphok és matrixok, Mat. Fiz. Lapok 38 (1931) 116–119.
[100] T.C. Koopmans, M.J. Beckmann, Assignment problems and the location of economic activities, Econometrica 25 (1957) 53–76.
[101] J. Krarup, P.M. Pruzan, Computer-aided layout design, Math. Programming Study 9 (1978) 75–94.
[102] G. Laporte, H. Mercure, Balancing hydraulic turbine runners: a quadratic assignment problem, European J. Oper. Res. 35 (1988) 378–382.
[103] E.L. Lawler, The quadratic assignment problem, Management Sci. 9 (1963) 586–599.
[104] A.J. Lazarus, Certain expected values in the random assignment problem, Oper. Res. Lett. 14 (1993) 207–214.
[105] Y. Lee, J.B. Orlin, On very large scale assignment problems, in: W.W. Hager, D.W. Hearn, P.M. Pardalos (Eds.), Large Scale Optimization: State of the Art, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994, pp. 206–244.
[106] J.L. Lewandowski, J.W.S. Liu, C.L. Liu, SS/TDMA time slot assignment with restricted switching modes, IEEE Trans. Comm. COM-31 (1983) 149–154.
[107] Y. Li, P.M. Pardalos, Generating quadratic assignment test problems with known optimal permutations, Comput. Optim. Appl. 1 (1992) 163–184.
[108] D. Magos, Tabu search for the planar three-index assignment problem, J. Global Optim. 8 (1996) 35–48.

[109] D. Magos, P. Miliotis, An algorithm for the planar three-index assignment problem, European J. Oper. Res. 77 (1994) 141–153.
[110] S. Martello, W.R. Pulleyblank, P. Toth, D. de Werra, Balanced optimization problems, Oper. Res. Lett. 3 (1984) 275–278.
[111] A. Marzetta, A. Brüngger, A dynamic programming bound for the quadratic assignment problem, in: Computing and Combinatorics: Fifth Annual International Conference COCOON'99, Lecture Notes in Computer Science, Vol. 1627, Springer, Heidelberg, 1999, pp. 339–348.
[112] E.J. McCormick, Human Factors Engineering, McGraw-Hill, New York, 1970.
[113] M. Mézard, G. Parisi, On the solution of the random link matching problems, J. Phys. 48 (1987) 1451–1459.
[114] J. Mosevich, Balancing hydraulic turbine runners—a discrete combinatorial optimization problem, European J. Oper. Res. 26 (1986) 202–204.
[115] K. Mulmuley, U.V. Vazirani, V.V. Vazirani, Matching is as easy as matrix inversion, Combinatorica 7 (1987) 105–113.
[116] R. Murphey, P.M. Pardalos, L.S. Pitsoulis, A GRASP for the multitarget multisensor tracking problem, in: P.M. Pardalos, D.-Z. Du (Eds.), Network Design: Connectivity and Facilities Location, DIMACS Series on Discrete Mathematics and Theoretical Computer Science, Vol. 40, AMS, Providence, RI, 1998, pp. 277–302.
[117] R. Murphey, P.M. Pardalos, L.S. Pitsoulis, A Parallel GRASP for the Data Association Multidimensional Assignment Problem, in: Parallel Processing of Discrete Problems, The IMA Volumes in Mathematics and its Applications, Vol. 106, Springer, Berlin, 1998, pp. 159–180.
[118] K.A. Murthy, P. Pardalos, Y. Li, A local search algorithm for the quadratic assignment problem, Informatica 3 (1992) 524–538.
[119] B. Neng, Zur Erstellung von optimalen Triebfahrzeugplänen, Z. Oper. Res. 25 (1981) B159–B185.
[120] B. Olin, Asymptotic Properties of Random Assignment Problems, Ph.D. Thesis, Division of Optimization and Systems Theory, Department of Mathematics, Royal Institute of Technology, Stockholm, 1992.
[121] M.W. Padberg, M.P. Rijal, Location, Scheduling, Design and Integer Programming, Kluwer Academic Publishers, Boston, 1996.
[122] E.S. Page, A note on assignment problems, Comput. J. 6 (1963) 241–243.
[123] C.H. Papadimitriou, D. Wolfe, The complexity of facets resolved, Proceedings of the 25th Annual IEEE Symposium on the Foundations of Computer Science (FOCS), 1985, pp. 74–78.
[124] P.M. Pardalos, F. Rendl, H. Wolkowicz, The quadratic assignment problem: a survey and recent developments, in: P.M. Pardalos, H. Wolkowicz (Eds.), Quadratic Assignment and Related Problems, DIMACS Series on Discrete Mathematics and Theoretical Computer Science, Vol. 16, AMS, Providence, RI, 1994, pp. 1–42.
[125] U. Pferschy, The random linear bottleneck assignment problem, RAIRO Oper. Res. 30 (1996) 127–142.
[126] U. Pferschy, Solution methods and computational investigations for the linear bottleneck assignment problem, Computing 59 (1997) 237–258.
[127] W.P. Pierskalla, The multidimensional assignment problem, Oper. Res. 16 (1968) 422–431.
[128] A.B. Poore, Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking, Comput. Optim. Appl. 3 (1994) 27–54.
[129] A.B. Poore, A numerical study of some data association problems arising in multitarget tracking, in: W.W. Hager, D.W. Hearn, P.M. Pardalos (Eds.), Large Scale Optimization: State of the Art, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994, pp. 339–361.
[130] A.B. Poore, N. Rijavec, Partitioning multiple data sets: multidimensional assignments and Lagrangian relaxation, in: P.M. Pardalos, H. Wolkowicz (Eds.), Quadratic Assignment and Related Problems, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 16, AMS, Providence, RI, 1994, pp. 317–342.
[131] A.B. Poore, N. Rijavec, M. Liggins, V. Vannicola, Data association problems posed as multidimensional assignment problems: problem formulation, in: O.E. Drummond (Ed.), Signal and Data Processing of Small Targets, SPIE, Bellingham, WA, 1993, pp. 552–561.
[132] A.B. Poore, A.J. Robertson III, A new Lagrangean relaxation based algorithm for a class of multidimensional assignment problems, Comput. Optim. Appl. 8 (1997) 129–150.

[133] A.P. Punnen, K.P.K. Nair, Improved complexity bound for the maximum cardinality bottleneck bipartite matching problem, Discrete Appl. Math. 55 (1994) 91–93.
[134] J. Pusztaszeri, P.E. Rensing, T.M. Liebling, Tracking elementary particles near their primary vertex: a combinatorial approach, J. Global Optim. 16 (1995) 422–431.
[135] L. Qi, E. Balas, G. Gwan, A new facet class and a polyhedral method for the three-index assignment problem, in: D.-Z. Du (Ed.), Advances in Optimization, Kluwer Academic Publishers, Dordrecht, 1994, pp. 256–274.
[136] M. Queyranne, Performance ratio of heuristics for triangle inequality quadratic assignment problems, Oper. Res. Lett. 4 (1986) 231–234.
[137] F. Rendl, On the complexity of decomposing matrices arising in satellite communication, Oper. Res. Lett. 4 (1985) 5–8.
[138] F. Rendl, H. Wolkowicz, Applications of parametric programming and eigenvalue maximization to the quadratic assignment problem, Math. Programming 53 (1992) 63–78.
[139] W.T. Rhee, Stochastic analysis of the quadratic assignment problem, Math. Oper. Res. 16 (1991) 223–239.
[140] C. Roucairol, A reduction method for quadratic assignment problems, Oper. Res. Verfahren 32 (1979) 183–187.
[141] S. Sahni, T. Gonzalez, P-complete approximation problems, J. Assoc. Comput. Mach. 23 (1976) 555–565.
[142] J.T. Schwarz, Fast probabilistic algorithms for verification of polynomial identities, J. ACM 27 (1980) 701–717.
[143] J. Skorin-Kapov, Tabu search applied to the quadratic assignment problem, ORSA J. Comput. 2 (1990) 33–45.
[144] W. Szpankowski, Combinatorial optimization problems for which almost every algorithm is asymptotically optimal!, Optimization 33 (1995) 359–367.
[145] L. Steinberg, The backboard wiring problem: a placement algorithm, SIAM Rev. 3 (1961) 37–50.
[146] E. Taillard, Robust tabu search for the quadratic assignment problem, Parallel Comput. 17 (1991) 443–455.
[147] R.E. Tarjan, Data Structures and Network Algorithms, SIAM, Philadelphia, PA, 1983.
[148] W.T. Tutte, The factorization of linear graphs, J. London Math. Soc. 22 (1947) 107–111.
[149] I. Ugi, J. Bauer, J. Friedrich, J. Gasteiger, C. Jochum, W. Schubert, Neue Anwendungsgebiete für Computer in der Chemie, Angew. Chem. 91 (1979) 99–111.
[150] M. Vlach, Branch and bound method for the three-index assignment problem, Ekonom.-Mat. Obzor 12 (1967) 181–191.
[151] D.W. Walkup, On the expected value of a random assignment problem, SIAM J. Comput. 8 (1979) 440–442.
[152] D.W. Walkup, Matching in random regular bipartite digraphs, Discrete Math. 31 (1980) 59–64.
[153] M.R. Wilhelm, T.L. Ward, Solving quadratic assignment problems by simulated annealing, IEEE Trans. 19 (1987) 107–119.
[154] Q. Zhao, S.E. Karisch, F. Rendl, H. Wolkowicz, Semidefinite relaxations for the quadratic assignment problem, J. Combin. Optim. 2 (1998) 71–109.